Final Project

Air Quality Dataset

Prepared by: Michael Trinh & Mayra Esparza

BUS2-193

May 11, 2023

Part 1 & 2 Introduction of Dataset

Within an Italian city, there was a dataset collection of hourly average air quality responses. The data was collected from a field within a populated area at road level over a one year time span, from the date March 2004 to February 2005. By using an Air Quality Chemical Multisensor Device, the data was separated into what different metal oxide chemical observations were found, CO, Non Metanic Hydrocarbons, Benzene, Total Nitrogen Oxides (NOx) and Nitrogen Dioxide (NO2). The dataset of 9358 recordings includes the date, time, and hourly averages of each chemical looked for in each individual data point collected.

Our chosen dataset includes 14 variables when input into R. Below are included each variable type that will be found.

Variable	Description
Date	Tells us when the data was recorded
Time	The exact time the data was collected
CO(GT)	Averaged concentration of Carbon Monoxide in mg per cubic meter - CO is colorless, odorless gas that is harmful when inhaled,
PT08.S1(CO)	Hourly average sensor response of Tin Oxide
NMHC(GT)	Hourly Average measurement of Non methane Hydrocarbons concentration in micrograms / cubic meters. - Important reactive gasses that play a key role in production and destruction of the ozone troposphere.
C6H6	Hourly Average measurement of Benzene concentration in micrograms / cubic meters
PT08.S2 (NMHC) (Titania)	Hourly average sensor response
NOx(GT)	True Hourly average NOx concentration in parts per billion - When in the presence of NMHC's, the reaction helps the formation of tropospheric ozone and stratospheric H2
PT08.S3(NOx)	Hourly average measurement of Tungsten oxide (NOx targeted) - Contribute to the formation of tropospheric ozone and stratosphere

NO2	Hourly Average of Nitrous DiOxide in micrograms / cubic meters
PT08.S4	Hourly Average sensor response (NO2 Targeted)
PT08.S5(O3)	Hourly average sensor response of Tungsten oxide (O3 Targeted)
T	Temperature in celsius
RH	Relative Humidity
AH	Absolute Humidity

Figure 0. Attribute Information

Using this dataset, we want to observe how different the air quality was throughout the year. By inputting the data in R, we can make charts to observe if we can find any correlation between variables, any outliers, and maybe see if different times or dates affect the responses. With the data, we want to be able to achieve a linear regression to show if there are any trends throughout the year. Our group decided on using temperature as our independent variable to report our findings being affected by temperature.

The data that was chosen comes from a main road in an urban area within an Italian city. The reason for the collection by the original collectors was to observe the pollutant levels in the area with a concentration of Co, No_x, NO_2, and benzene.

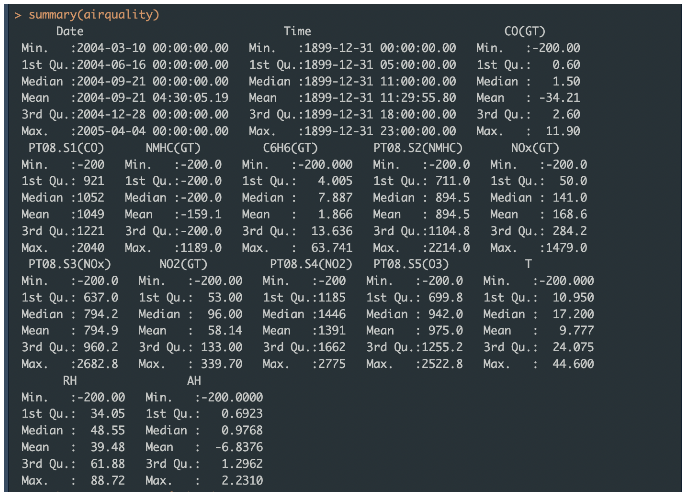
Part 3 Preliminary Analysis

>Str(airquality)

```
> str(airquality)
tibble [9,357 x 15] (S3: tbl_df/tbl/data.frame)
 $ Date         : POSIXct[1:9357], format: "2004-03-10" "2004-03-10" ...
 $ Time         : POSIXct[1:9357], format: "1899-12-31 18:00:00" "1899-12-31 19:00:00" ...
 $ CO(GT)       : num [1:9357] 2.6 2 2.2 2.2 1.6 1.2 1.2 1 0.9 0.6 ...
 $ PT08.S1(CO)  : num [1:9357] 1360 1292 1402 1376 1272 ...
 $ NMHC(GT)     : num [1:9357] 150 112 88 80 51 38 31 31 24 19 ...
 $ C6H6(GT)     : num [1:9357] 11.88 9.4 9 9.23 6.52 ...
 $ PT08.S2(NMHC): num [1:9357] 1046 955 939 948 836 ...
 $ NOx(GT)      : num [1:9357] 166 103 131 172 131 89 62 62 45 -200 ...
 $ PT08.S3(NOx) : num [1:9357] 1056 1174 1140 1092 1205 ...
 $ NO2(GT)      : num [1:9357] 113 92 114 122 116 96 77 76 60 -200 ...
 $ PT08.S4(NO2) : num [1:9357] 1692 1559 1554 1584 1490 ...
 $ PT08.S5(O3)  : num [1:9357] 1268 972 1074 1203 1110 ...
 $ T            : num [1:9357] 13.6 13.3 11.9 11 11.2 ...
 $ RH           : num [1:9357] 48.9 47.7 54 60 59.6 ...
 $ AH           : num [1:9357] 0.758 0.725 0.75 0.787 0.789 ...
>
```

Figure 1. Structure analysis of the data set

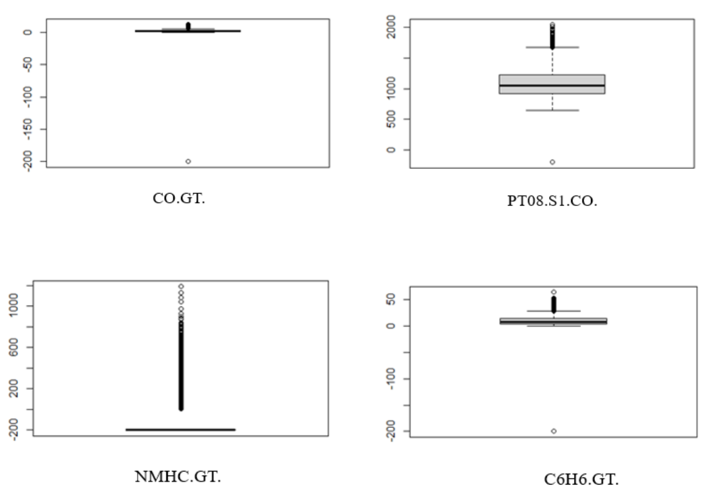
In figure 1, all of our variables are numerical data besides Date and Time which is using a date time format.



```
> summary(airquality)
      Date                          Time                          CO(GT)
 Min.   :2004-03-10 00:00:00.00   Min.   :1899-12-31 00:00:00.00   Min.   :-200.00
 1st Qu.:2004-06-16 00:00:00.00   1st Qu.:1899-12-31 05:00:00.00   1st Qu.:   0.60
 Median :2004-09-21 00:00:00.00   Median :1899-12-31 11:00:00.00   Median :   1.50
 Mean   :2004-09-21 04:30:05.19   Mean   :1899-12-31 11:29:55.80   Mean   : -34.21
 3rd Qu.:2004-12-28 00:00:00.00   3rd Qu.:1899-12-31 18:00:00.00   3rd Qu.:   2.60
 Max.   :2005-04-04 00:00:00.00   Max.   :1899-12-31 23:00:00.00   Max.   :  11.90
  PT08.S1(CO)     NMHC(GT)        C6H6(GT)        PT08.S2(NMHC)     NOx(GT)
 Min.   :-200   Min.   :-200.0   Min.   :-200.000   Min.   :-200.0   Min.   :-200.0
 1st Qu.: 921   1st Qu.:-200.0   1st Qu.:   4.005   1st Qu.: 711.0   1st Qu.:  50.0
 Median :1052   Median :-200.0   Median :   7.887   Median : 894.5   Median : 141.0
 Mean   :1049   Mean   :-159.1   Mean   :   1.866   Mean   : 894.5   Mean   : 168.6
 3rd Qu.:1221   3rd Qu.:-200.0   3rd Qu.:  13.636   3rd Qu.:1104.8   3rd Qu.: 284.2
 Max.   :2040   Max.   :1189.0   Max.   :  63.741   Max.   :2214.0   Max.   :1479.0
  PT08.S3(NOx)      NO2(GT)        PT08.S4(NO2)   PT08.S5(O3)          T
 Min.   :-200.0   Min.   :-200.00   Min.   :-200   Min.   :-200.0   Min.   :-200.000
 1st Qu.: 637.0   1st Qu.:  53.00   1st Qu.:1185   1st Qu.: 699.8   1st Qu.:  10.950
 Median : 794.2   Median :  96.00   Median :1446   Median : 942.0   Median :  17.200
 Mean   : 794.9   Mean   :  58.14   Mean   :1391   Mean   : 975.0   Mean   :   9.777
 3rd Qu.: 960.2   3rd Qu.: 133.00   3rd Qu.:1662   3rd Qu.:1255.2   3rd Qu.:  24.075
 Max.   :2682.8   Max.   : 339.70   Max.   :2775   Max.   :2522.8   Max.   :  44.600
       RH               AH
 Min.   :-200.00   Min.   :-200.0000
 1st Qu.:  34.05   1st Qu.:   0.6923
 Median :  48.55   Median :   0.9768
 Mean   :  39.48   Mean   :  -6.8376
 3rd Qu.:  61.88   3rd Qu.:   1.2962
 Max.   :  88.72   Max.   :   2.2310
```

Figure 2. Summary Statistics of Air Quality dataset

In Figure 2. We can see most of our variables have a minimum quantity of -200. Some things to note, each of our variables are in different measurement values. CO(GT) is mg/gm^3, NMHC(GT) and NO2(GT) are in microg/m^3, T (temperature) in °C, RH and possibly AH in percentage, and the rest are measured in hourly average sensor responses.



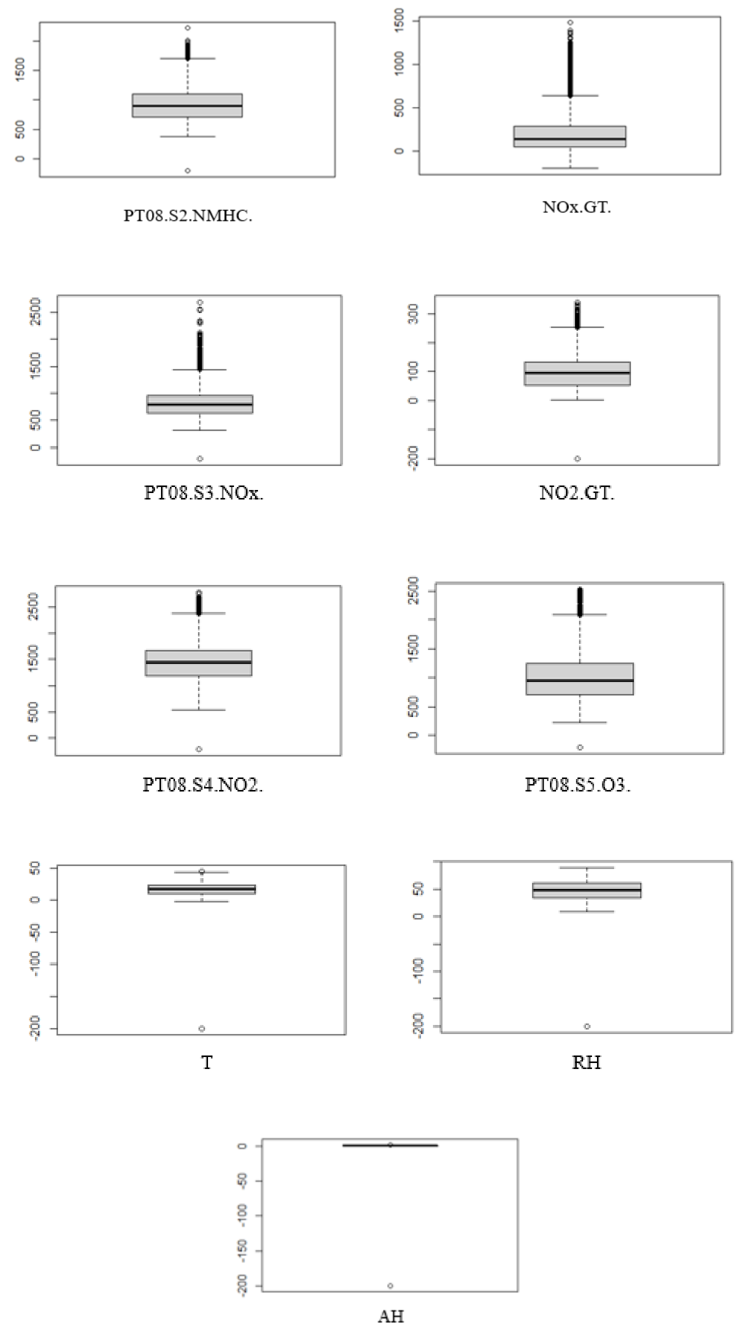
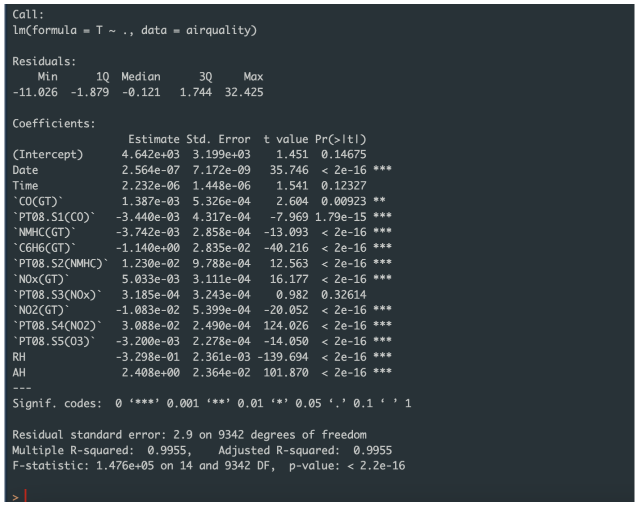


Fig. 3 Boxplots of each variable

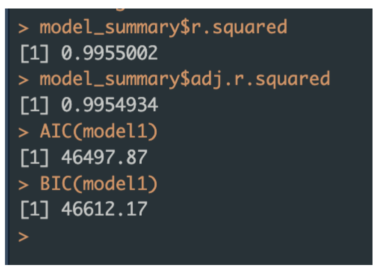
With the boxplot() function, we use it to observe how the data is spread out with date and time not included. This way, we can identify outliers within our dataset by a visual representation. It seems as if every variable has at least one outlier. While looking through each, we can see that NCHC.GT, NOx.GT, and PT08.S3.NOx have the most outliers when compared to the rest. By having this many outliers, our results can be affected by this and possibly give us inaccurate results.



```
Call:
lm(formula = T ~ ., data = airquality)

Residuals:
    Min      1Q  Median      3Q     Max
-11.026  -1.879  -0.121   1.744  32.425

Coefficients:
                  Estimate Std. Error  t value Pr(>|t|)
(Intercept)      4.642e+03  3.199e+03    1.451  0.14675
Date             2.564e-07  7.172e-09   35.746  < 2e-16 ***
Time             2.232e-06  1.448e-06    1.541  0.12327
`CO(GT)`         1.387e-03  5.326e-04    2.604  0.00923 **
`PT08.S1(CO)`   -3.440e-03  4.317e-04   -7.969 1.79e-15 ***
`NMHC(GT)`      -3.742e-03  2.858e-04  -13.093  < 2e-16 ***
`C6H6(GT)`      -1.140e+00  2.835e-02  -40.216  < 2e-16 ***
`PT08.S2(NMHC)`  1.230e-02  9.788e-04   12.563  < 2e-16 ***
`NOx(GT)`        5.033e-03  3.111e-04   16.177  < 2e-16 ***
`PT08.S3(NOx)`   3.185e-04  3.243e-04    0.982  0.32614
`NO2(GT)`       -1.083e-02  5.399e-04  -20.052  < 2e-16 ***
`PT08.S4(NO2)`   3.088e-02  2.490e-04  124.026  < 2e-16 ***
`PT08.S5(O3)`   -3.200e-03  2.278e-04  -14.050  < 2e-16 ***
RH              -3.298e-01  2.361e-03 -139.694  < 2e-16 ***
AH               2.408e+00  2.364e-02  101.870  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.9 on 9342 degrees of freedom
Multiple R-squared:  0.9955,    Adjusted R-squared:  0.9955
F-statistic: 1.476e+05 on 14 and 9342 DF,  p-value: < 2.2e-16

>
```

Fig 4. Regression Model

We ran a linear regression model and here we can see Date, CO, PT08.S1, NHMC,C6H6, PT08.S2, NOx, NO2, PT08.S4, PT08.S5, RH, AH are significant.



```
> model_summary$r.squared
[1] 0.9955002
> model_summary$adj.r.squared
[1] 0.9954934
> AIC(model1)
[1] 46497.87
> BIC(model1)
[1] 46612.17
>
```

Fig 5. Checking goodness of fit

When checking our model fit, we had an r-squared of 0.9955002, and adjusted r-squared of 0.9954934, AIC of 46497.87 and BIC of 46612.17. Our r-squared and adjusted r-squared values are both good but our AIC and BIC is telling us we have a lot of information lost. This could be from our variable “PT08.S3(NOx)” being insignificant.

Part 4 Proposed Work

From this data, our next step will be to separate our dataset into a training and testing set. By reviewing the outputs, we will see if the outliers are similar to the raw data in both the training and testing datasets. We will continue our project with using the training dataset to fit a new linear regression model. The model will show us if any variables are significant for our data. Using the training model, we will use the goodness of fit to then find the MSE, adjusted R^2, AIC and BIC. With this we can conclude if our results are reasonable. We also plan to do an out-of-sample prediction with the testing dataset and compare to our training set to conclude which is the best fitted model.

Our group still needs to overview which new data mining set we will use for our project but will be deciding soon. As of now, we are reviewing our dataset to be sure we can find suitable results for our end goal of identifying possible trends within our datasets. With the results we will prepare a presentation explaining our findings. This will show what results were achieved by using new data mining techniques and give us what we hoped to achieve from this project.

Part 5 Corrections

After reviewing our results from our proposed section, we decided to delete some of the variables from the model. The variables removed are “Date”, “Time”, and “NMHC(GT)”for our final model building. The reason for removing date and time is that both variables do not give any significant measurements to help predict a linear regression for our project. We also decided to remove “NMHC(GT)” since much of the data was not measured, giving us a lot of blank data points that can affect our end results. Another issue we ran across was that many data points were set at -200 which were used as no data recorded for those points. Each -200 were converted to NA and then replaced to an average measurement in their original column to help fill in the blank spots for our data.

Part 6 Model Building

To begin our model building, we split our data into 80% for a training data set and 20% for testing data set. We fit the training data set to a linear regression model with the response variable being temperature (T). Below is the summary output of our training model.

```
> summary(air1)

Call:
lm(formula = T ~ ., data = air.train)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.3190 -1.3473 -0.1487  1.0938 13.3906 

Coefficients:
                Estimate Std. Error  t value Pr(>|t|)    
(Intercept)    8.902e+00  6.703e-01   13.281  < 2e-16 ***
CO.GT.        -1.983e-01  4.389e-02   -4.518 6.34e-06 ***
PT08.S1.CO.   -2.012e-05  3.528e-04   -0.057   0.9545    
C6H6.GT.      -3.870e-01  2.295e-02  -16.864  < 2e-16 ***
PT08.S2.NMHC.  1.057e-02  7.938e-04   13.310  < 2e-16 ***
NOx.GT.        2.650e-03  3.173e-04    8.353  < 2e-16 ***
PT08.S3.NOx.   3.367e-04  2.442e-04    1.378   0.1681    
NO2.GT.       -2.337e-03  1.152e-03   -2.029   0.0425 *  
PT08.S4.NO2.   5.722e-03  2.779e-04   20.590  < 2e-16 ***
PT08.S5.O3.   -2.691e-03  1.932e-04  -13.925  < 2e-16 ***
RH            -3.388e-01  2.059e-03 -164.548  < 2e-16 ***
AH             1.392e+01  1.418e-01   98.142  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.257 on 7564 degrees of freedom
Multiple R-squared:  0.9311,	Adjusted R-squared:  0.931 
F-statistic:  9296 on 11 and 7564 DF,  p-value: < 2.2e-16
```

Fig. 6 Logistic Regression of Air Quality Training Set

Here we can see each variable besides PT08.S1.CO, PT08.S3.N0X, and NO2.GT are highly significant to our response variable temperature (T).

Next we want to do the in-sample model evaluation using the training data and then compare with the out-of-sample model that uses the testing data set.

```
> #Evaluating Model Fitness
> #In-sample with training data
> #MSE
> air_summary <- (summary(air1))
> (air_summary$sigma)^2
[1] 5.094892
> #R^2 of model
> air_summary$r.squared
[1] 0.9311228
> #Adjusted R^2
> air_summary$adj.r.squared
[1] 0.9310226
> AIC(air1)
[1] 33849.28
> BIC(air1)
[1] 33939.41
```

Fig. 7 Model Evaluations

Here are the MSE, R^2, adjusted R^2, AIC, and BIC of our in-sample model. For the out-of-sample, we use the testing data set and predict the performance on future data. With the predict() function, we can get the MSE and MAE of the model.

```
> #MSE
> mean((pi - air.test$T)^2)
[1] 309.4123
> #MAE
> mean(abs(pi - air.test$T))
[1] 15.35963
```

Fig. 8 MSE and MAE

We also want to look at variable selection by using regression trees with our model. By using the rpart() function, we get a similar output model just as the lm() function would give us.

```
> air_rpart
n= 7576

node), split, n, deviance, yval
      * denotes terminal node

 1) root 7576 559513.700 18.282470
   2) AH< 0.8333 2406 108048.100 11.043020
     4) RH>=31.85 1901  37359.740  8.813624
       8) AH< 0.5814 840   8629.403  6.322024 *
       9) AH>=0.5814 1061  19386.980 10.786240
        18) RH>=52.15 640   4139.600  8.037500 *
        19) RH< 52.15 421   3060.840 14.964850 *
     5) RH< 31.85 505  25673.090 19.435250
      10) AH< 0.49305 212   5213.570 13.428300 *
      11) AH>=0.49305 293   7274.920 23.781570 *
   3) AH>=0.8333 5170 266684.800 21.651550
     6) RH>=39.95 3812  87017.060 18.550710
      12) AH< 1.2964 2402  30136.160 16.137810
        24) RH>=57.65 1061   5821.148 13.113100 *
        25) RH< 57.65 1341   6927.956 18.530960 *
      13) AH>=1.2964 1410  19072.610 22.661210
        26) RH>=60.65 613   4195.823 19.654000 *
        27) RH< 60.65 797   5069.508 24.974150 *
     7) RH< 39.95 1358  40126.730 30.355820
      14) RH>=27.85 772  13437.250 27.250520
        28) AH< 1.257 449   2818.552 24.486860 *
        29) AH>=1.257 323   2422.171 31.092260 *
      15) RH< 27.85 586   9437.999 34.446760 *
```

Fig. 9 rpart Model

Here we use our training data set to build into our new rpart model. By having this output, the next step is to build the tree using the prp() to draw the tree for use. What we noticed is our outcome only gives the variables relative and absolute humidity. Looking at our tree build, it reads by comparing the values we want to test through the trees but only with absolute and relative humidity.

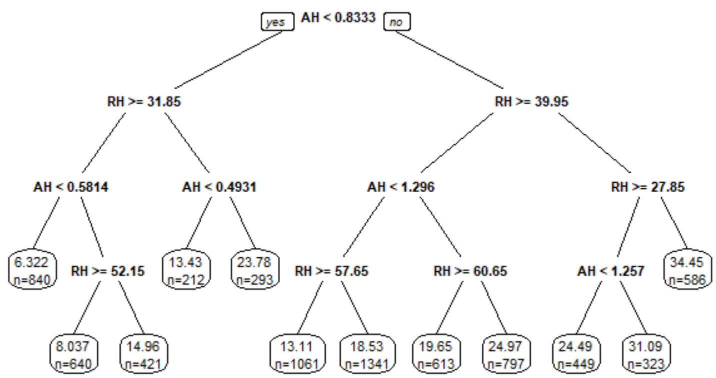


Fig. 10 Regression Tree

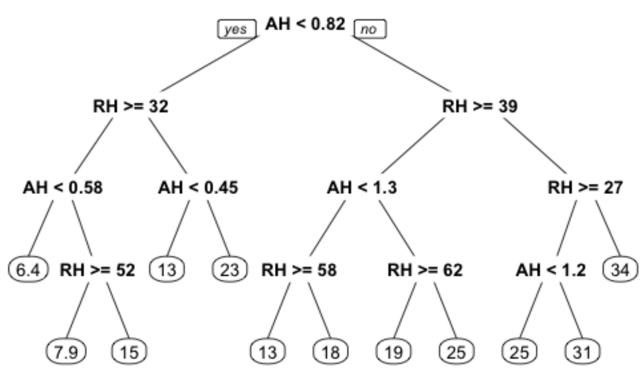


Fig. 10-1 Pruned Tree

Next, we want to compare the in and out-of-sample performance of the regression tree. For the in-sample, we used the predict() function to help report the MSE of the model. The in-sample prediction uses the training data set of the rpart model.

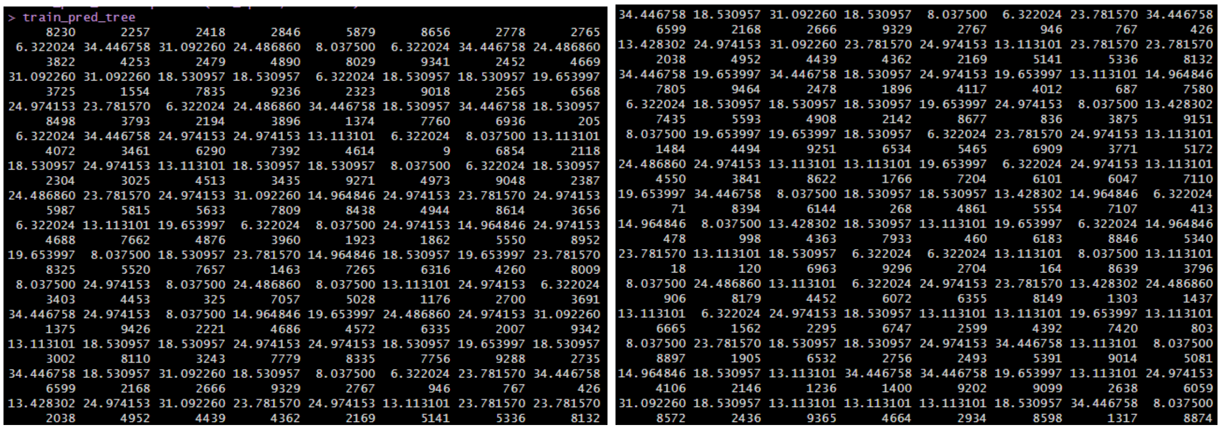


Fig. 11 Training Data Prediction Model

The out-of-sample prediction snippet is below. We did not show everything since there are many data points that it does cut off at a certain point. This prediction uses the rpart model with an additional argument of the testing data set. To compare, we use the out-of-sample performance and the linear regression model to get the MSE.

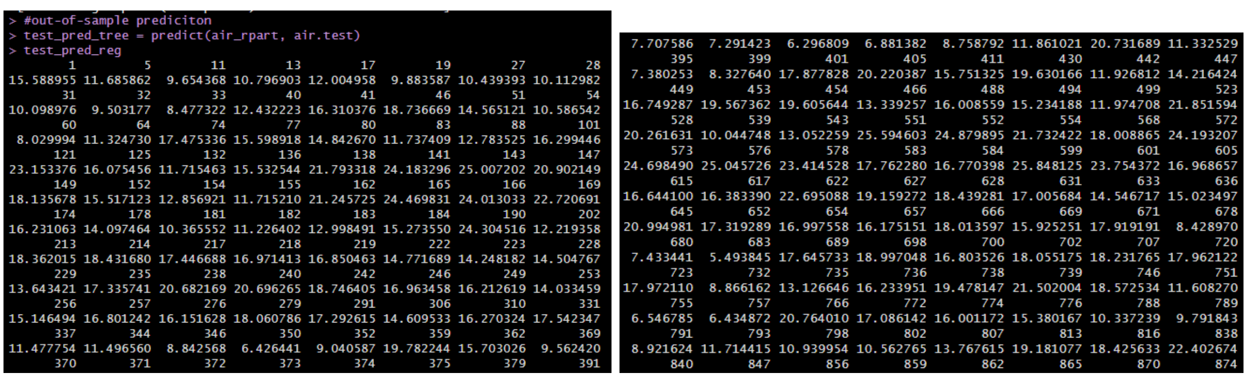


Fig. 12 Testing Data Prediction Model

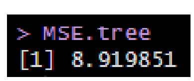


Fig. 13 MSE

Neural Network

The new data mining tool our group decided to use was Neural Network. This tool can be used to improve the accuracy of the data over time using our training data. Something to note, the outputs are suspicious that our data returned a very small MSE although we changed our code to the parameters required to mirror our code. We are not really sure what happened to give us small MSEs, but these were the results given for the Neural Network code.

Since we already have our data split, we use our air.train set to build our model. For this section, we included what code was used. The column was changed to -12 since we have 12 variables and the highest temperature is 44.6 which our data was divided and multiplied by as well.

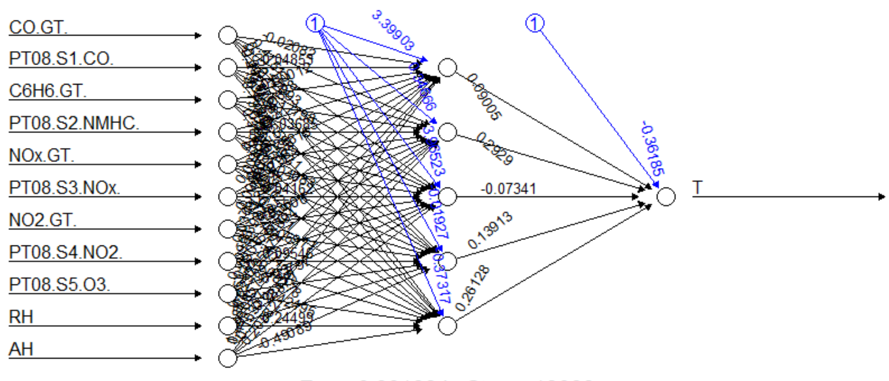


Fig. 14 Neural Network Model 1 Hidden Layer

Now we want to do a prediction on the testing sample set to get the MSE to compare our testing data set with the predicted set. We get a MSE of 0.0011 which we can confidently say that our data points from both sets are close to each other.

```
> air.pred1<- compute(air.ann1, air.test)
> air.pred1<- air.pred1$net.result*44.6
> mean((air.test$T-air.pred1)^2)
[1] 0.001137391
```

Fig. 15 MSE of Prediction and Testing Data

Let’s try with another hidden layer in our network.

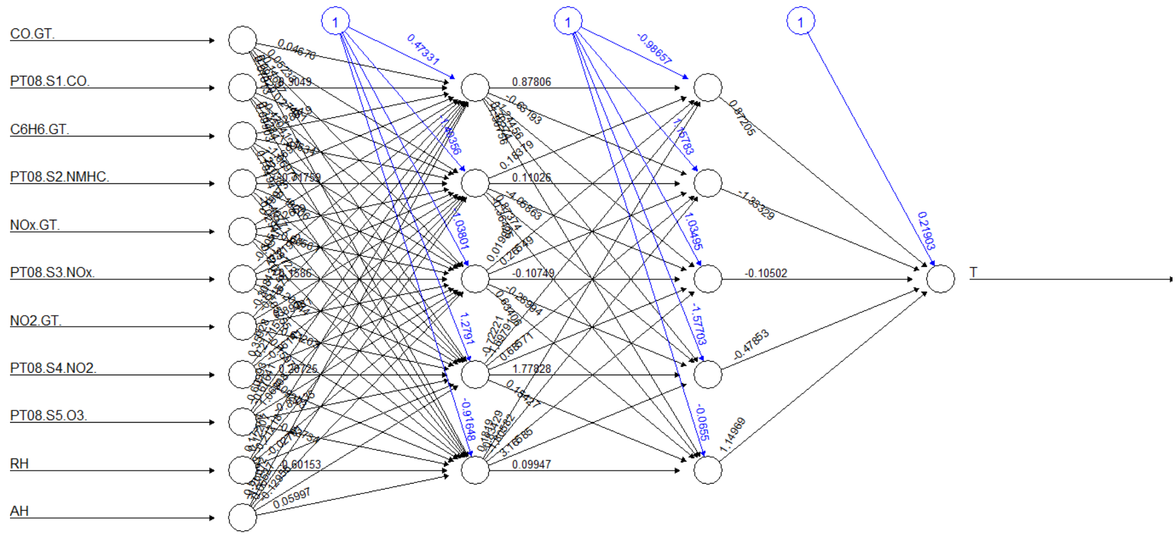


Fig. 16 Neural Network Model 2 Hidden Layers

Our MSE for the second comparison with a second hidden layer gives us a number of 0.0046 which is very small. This could mean that our second comparison is even closer to being a similar data point when compared to the first.

```
> air.pred2<- compute(air.ann2, air.test)
> air.pred2<- air.pred2$net.result*44.6
> mean((air.test$T-air.pred2)^2)
[1] 0.004584932
```

Fig. 17 MSE of Prediction2 and Testing Data

For both plots, the leftmost nodes are our raw data variables. Each arrow has a weight that seems to contribute to the next node it leads to. The blue arrows from the hidden nodes are bias weights that can shift a value closer to the output we would want. Last part we did was compare our air quality regression prediction with the air.test data to get the MSE. This model comparison is not the better model. The second one we compared using the air prediction and air test was the best model with the lowest MSE.

```
> mean((air.test$T-air.reg.pred)^2)
[1] 0.06887571
```

Fig. 18 MSE of Prediction Linear Regression and Testing

After comparing all 3 models, We can conclude the linear regression model is the best. Even though the neural network model has the lowest MSE, it is suspiciously too low. We cannot feel comfortable with choosing that model.

List of references

De Vito, S., Di Francia, G., Martinotto, L., Piga, M., & Massera, E. (2008, February 22). *Air Quality Data Set*. UCI Machine Learning Repository: Air Quality Data Set. Retrieved April 3, 2023, from https://archive.ics.uci.edu/ml/datasets/Air+Quality#

De Vito, S., Massera, E., Piga, M., Martinotto, L., & Di Francia, G. (2007, September 25). *On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario*. Sensors and Actuators B: Chemical. Retrieved April 3, 2023, from https://www.sciencedirect.com/science/article/pii/S0925400507007691