# Project2

2023-04-14

**Title: Triple P (Pollution Predictor Platform)**

**Names: Vrishank Jannu, Adhi Karthik**

---

**Part 1: Introduction**

For this project we decided to analyze the predictor variables of CMAQ and aod to witness if there was a relationship between the response variable of PM 2.5 concentration. Background information on this data suggests that the EPA avg level of PM 2.5 concentration should be less than 12 micrograms / meters cubed, however many counties in the U.S. were still above the standard and had to find ways to lower their PM2.5 levels. With that being said, an overview visualization of the P.M 2.5 concentration of America was plotted before wrangling the data into testing and training datasets which was then implemented into the 4 models.

The 4 modeling approaches choosen were linear, random forest, k-nearest, and poisson regression models. Linear and k-nearest models were all tried and tested models implemented in the duration of class and the poisson regression and the random forest model were tested for the specific dataset through use of tuning parameters that would hopefully give appropriate results. Furthermore, the predictor variables of CMAQ and aod were consistent across the project as these are the future cost effective methods to monitor air pollution on the ground, and we felt that modeling the data using these two predictors would provide significant evidence to discard ground based monitors.

Scatterplots for each visualization were created in order to view the goodness of fit for each model. Residual plots and interaction plots were also included in some models to try and improve the prediction performance. Lastly, we felt that the linear, random forest, and k-nearest models should have similar, low RMSE values whereas the poisson model will have a higher number.

First, let's load the data and any packages we will need. Let's also view what the data looks like in a table format.

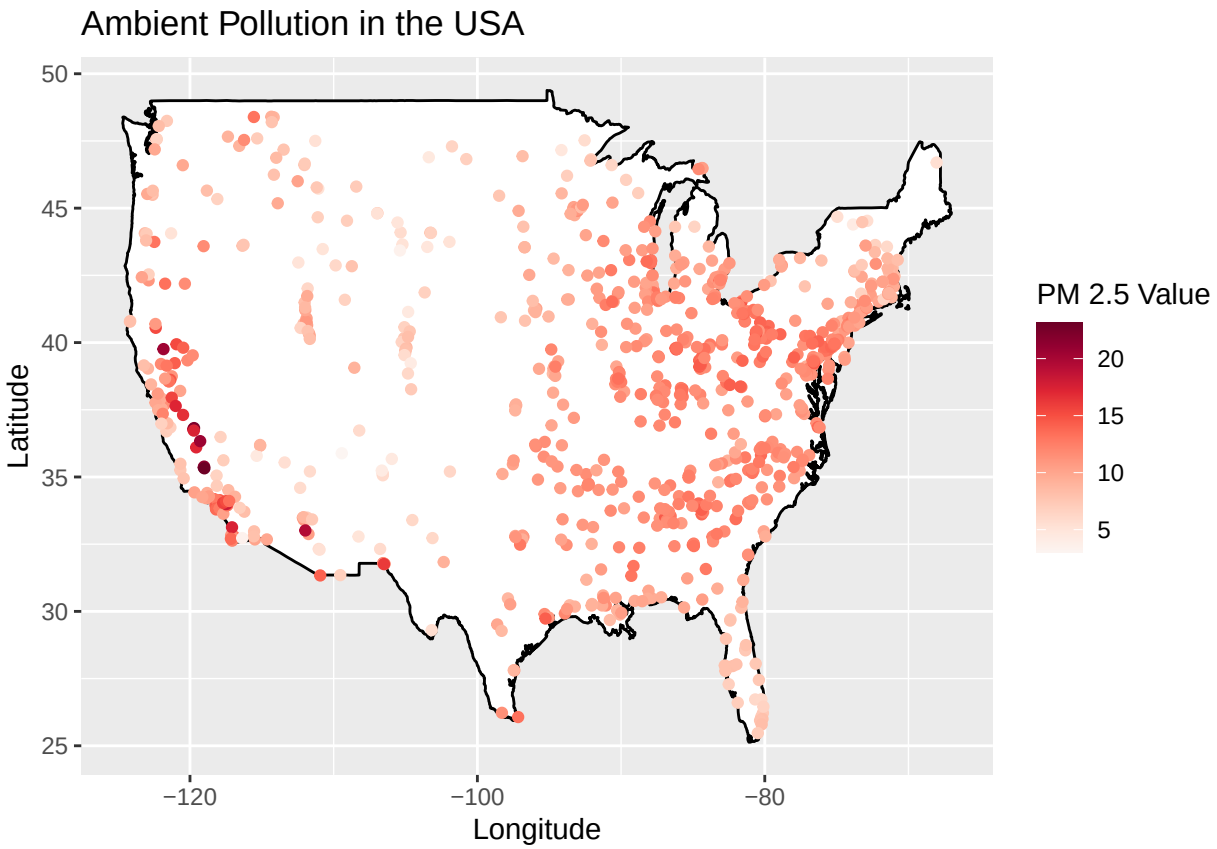
```
library(tidyverse)
library(kknn)
library(plotROC)
library(tidymodels)
library(ggplot2)
library(maps)
library(caret)
library(ggplot2)
library(randomForest)
library(glmnet)
library(MASS)
library(magrittr)
library(colorspace)
```

```r
dat <- read_csv("https://github.com/rdpeng/stat322E_public/raw/main/data/pm25_data.csv.gz")
dat %>%
  sample_n(10) # Shows a sample of sample size 10 of the the dat dataset
```

```
## # A tibble: 10 x 50
##        id value  fips   lat    lon state        county city   CMAQ  zcta zcta_~1
##     <dbl> <dbl> <dbl> <dbl>  <dbl> <chr>        <chr>  <chr> <dbl> <dbl>   <dbl>
##  1 20091.  8.45 20091  38.8  -94.7 Kansas       Johns~ Not ~  8.03 66062  1.24e8
##  2 25009.  8.71 25009  42.5  -71.0 Massachuset~ Essex  Lynn   8.04  1904  1.17e7
##  3 38007.  4.57 38007  46.9 -103.  North Dakota Billi~ Not ~  2.76 58622  1.62e9
##  4  6025. 10.2   6025  32.7 -115.  California   Imper~ Cale~ 14.9  92231  1.48e8
##  5 36029. 10.7  36029  42.9  -78.8 New York     Erie   Buff~  9.75 14206  1.26e7
##  6 47065. 11.3  47065  35.2  -85.2 Tennessee    Hamil~ Sodd~  8.11 37379  2.74e8
##  7 17099. 10.6  17099  41.3  -89.0 Illinois     La Sa~ Ogle~  8.54 61348  9.87e7
##  8 41061.  6.72 41061  45.3 -118.  Oregon       Union  La G~  2.31 97850  7.49e8
##  9  8005.  7.18  8005  39.6 -105.  Colorado     Arapa~ Litt~  4.57 80120  2.15e7
## 10 16009.  9.05 16009  47.3 -117.  Idaho        Benew~ Not ~  5.97 83861  1.32e9
## # ... with 39 more variables: zcta_pop <dbl>, imp_a500 <dbl>, imp_a1000 <dbl>,
## #   imp_a5000 <dbl>, imp_a10000 <dbl>, imp_a15000 <dbl>, county_area <dbl>,
## #   county_pop <dbl>, log_dist_to_prisec <dbl>, log_pri_length_5000 <dbl>,
## #   log_pri_length_10000 <dbl>, log_pri_length_15000 <dbl>,
## #   log_pri_length_25000 <dbl>, log_prisec_length_500 <dbl>,
## #   log_prisec_length_1000 <dbl>, log_prisec_length_5000 <dbl>,
## #   log_prisec_length_10000 <dbl>, log_prisec_length_15000 <dbl>, ...
```

We will now take a look at where the PM2.5 monitors are located geographically. For this we can use the `maps` package in R which contains a US map overlay. To display the locations, we can plot the `lat` and `lon` (latitude and longitude) values across the x and y axis using `geom_point` from the `ggplot2` library. This plot tells us a lot about the data that we are working with. We can see that there is a much higher concentration of PM2.5 monitors in the East Coast because there are more cities and densely populated regions east of the Mississippi River.

```r
us_map <- map_data("usa")
# Create a ggplot object and add the map data as a polygon layer
ggplot() +
  geom_polygon(data = us_map, aes(x = long, y = lat, group = group),
               fill = "white", color = "black") +
  # Use the geom_polygon function to accurately represent the states in the US
  geom_point(data = dat, aes(x = lon, y = lat, color = value)) +
  # Add axis labels and a plot title
  labs(title = "Ambient Pollution in the USA",
       x = "Longitude", y = "Latitude",
       color = "PM 2.5 Value") +
       scale_color_continuous_sequential(palette = "Reds")
```




**Part 2: Wrangling**

```
# SPLITTING UP DATASET INTO REGIONS
east <- dat[dat$lon > -100, ]
west <- dat[dat$lon < -100, ]
north <- dat[dat$lat > 38, ]
south <- dat[dat$lat < 38, ]

dat <- dat %>%
mutate(CMAQ_aod = CMAQ * aod) # Create interaction term
```

In order to determine our "best and final model", we will first split the data into training and testing sets. We will use the same training and testing data for all four models to maintain consistency. Given that `CMAQ` and `aod` are considered to be among the most important predictors of air pollution, we will use these two values to perform four models: linear regression, poisson regression model, random forest regression, and the k-nearest-neighbors classification model.

## Linear Regression Model

```
# Data filtering
# Set testing and training data to evaluate in unbiased manner
set.seed(123)
```

```r
train_index <- createDataPartition(dat$value, p = 0.5, list = FALSE)
# Set the size of the training dataset to be 50% the filtered data
train <- dat[train_index, ]
test <- dat[-train_index, ]
# Model summary of data
lm_model <- lm(value ~ CMAQ + aod, data = train)
summary(lm_model)
```

```
##
## Call:
## lm(formula = value ~ CMAQ + aod, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8753 -1.1700 -0.0067  1.1348 11.8552
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.780952   0.347625  19.507  < 2e-16 ***
## CMAQ        0.329922   0.036123   9.133  < 2e-16 ***
## aod         0.030078   0.005684   5.292 1.92e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.178 on 437 degrees of freedom
## Multiple R-squared:  0.2564, Adjusted R-squared:  0.253
## F-statistic: 75.36 on 2 and 437 DF,  p-value: < 2.2e-16
```

```r
# Test predictions on data
predictions_lm <- predict(lm_model, newdata = test)
rmse_1 <- sqrt(mean((test$value - predictions_lm)^2))
#Store results for model in rmse_1
```

Created a simple linear regression model that showcases the relationship between the predictor variables CMAQ and aod and the response variable "value" that indicates concentration of PM 2.5. CMAQ is a predictor variable that simulates pollution in the atmosphere, and aod stands for "aerosol optical depth" that relates to the amount of pollution near the surface. After running the linear regression model the summary statistics showcase a very low p-value alongside a high F value which demonstrates that the predictor variables are very significant in affecting the concentration of PM 2.5.

## Linear Regression Performance Improvements

```r
# Set testing and training data to evaluate in unbiased manner
set.seed(123)
train_index <- createDataPartition(dat$value, p = 0.5, list = FALSE)
# Set the size of the training dataset to be 50% the filtered data
train_int <- dat[train_index, ]
test_int <- dat[-train_index, ]
# Create interaction term in both training and test datasets
train_int <- train_int %>% mutate(CMAQ_aod = CMAQ * aod)
test_int <- test_int %>% mutate(CMAQ_aod = CMAQ * aod)
```

```r
# Model Summary of data
lm_model_int <- lm(value ~ CMAQ + aod + CMAQ_aod, data = train_int)
summary(lm_model_int)
```

```
##
## Call:
## lm(formula = value ~ CMAQ + aod + CMAQ_aod, data = train_int)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2156 -1.1190  0.0348  1.0843 11.6563
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.715491   0.806746   5.845 9.93e-09 ***
## CMAQ         0.570461   0.092185   6.188 1.40e-09 ***
## aod          0.081550   0.019030   4.285 2.25e-05 ***
## CMAQ_aod    -0.005749   0.002030  -2.832  0.00484 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.161 on 436 degrees of freedom
## Multiple R-squared:  0.2699, Adjusted R-squared:  0.2648
## F-statistic: 53.72 on 3 and 436 DF,  p-value: < 2.2e-16
```

```r
# Test data predictions
test_predictions_lm <- predict(lm_model_int, newdata = test_int)
# root mean squared errors of linear model with interaction term
rmse_int <- sqrt(mean((test_int$value - test_predictions_lm)^2))
```

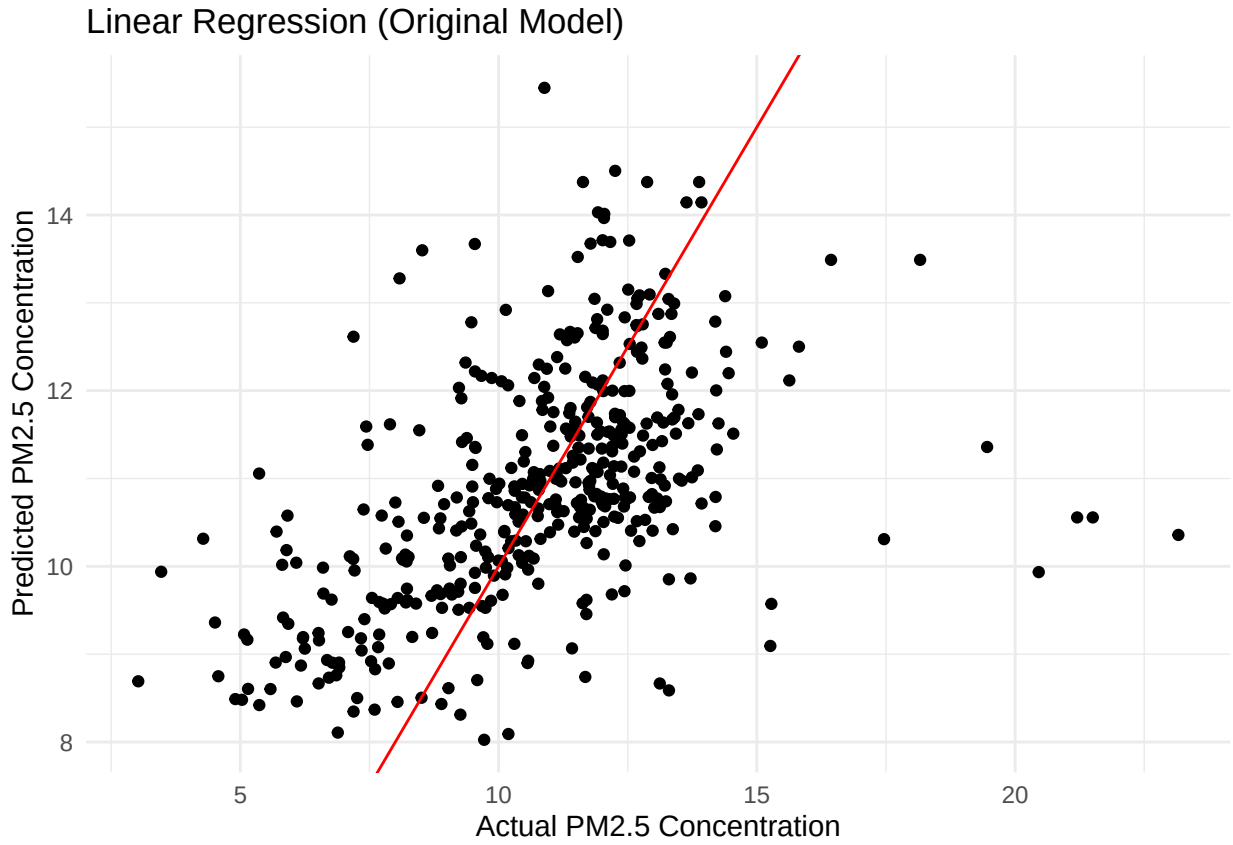
In order to further explore the performance of the linear model, a transformation of the predictor variables were done through taking the product of CMAQ and aod which is labelled as the CMAQ_aod variable. Since the estimate of the CMAQ_ado variable was negative (-0.005749) we can conclude that the effect of CMAQ decreases as aod decreases and vice versa. This can lead us to interpret that both predictor variables are significant and the impact of aod on pm2.5 depends on CMAQ and the impact of CMAQ on pm25 depends on aod.

### Exploratory Analysis

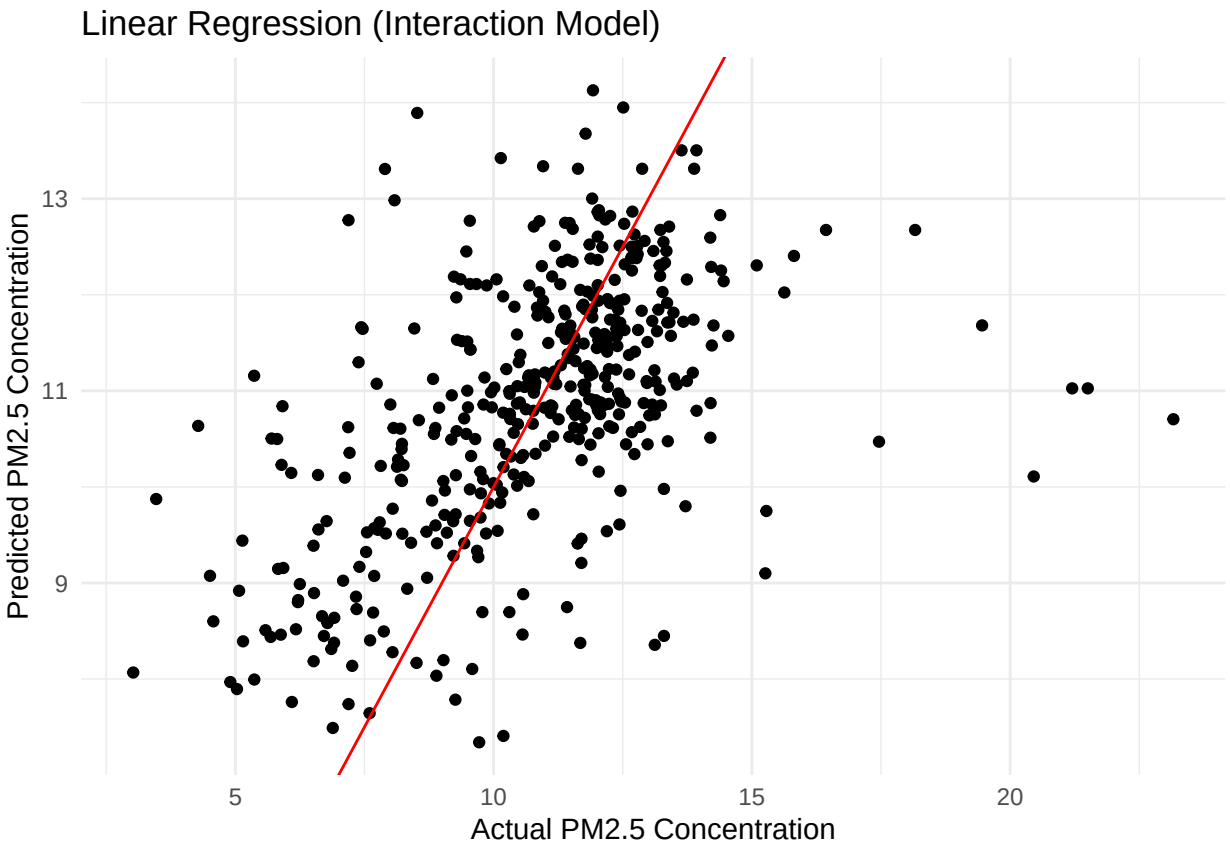
## Linear Regression Prediction Performance Visualization

```r
# Scatterplot of predicted vs actual values for original linear model
test_predictions_1 <- predict(lm_model, newdata = test)
# Creating a data frame to store the actual and predicted values
plot_data <- data.frame(actual = test$value, predicted = test_predictions_1)
# Creating the scatterplot for linear model
ggplot(data = plot_data, aes(x = actual, y = predicted)) +
geom_point() +
geom_abline(intercept = 0, slope = 1, color = "red") +
labs(title = "Linear Regression (Original Model)",
x = "Actual PM2.5 Concentration",
```

```
y = "Predicted PM2.5 Concentration") +
theme_minimal()
```




```
# Scatterplot of predicted vs actual values for linear model with an interaction
test_predictions_int <- predict(lm_model_int, newdata = test_int)
# Creating a data frame to store the actual and predicted values
plot_data_int <- data.frame(actual = test_int$value, predicted = test_predictions_int)
# Creating the scatterplot for linear model with interaction
ggplot(data = plot_data_int, aes(x = actual, y = predicted)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red") +
  labs(title = "Linear Regression (Interaction Model)",
       x = "Actual PM2.5 Concentration",
       y = "Predicted PM2.5 Concentration") +
  theme_minimal()
```




We can see from both the normal linear regression plot along with the linear regression plot with an interaction that the predicted values seem to have a similar fit to the linear regression model with an interaction. Knowing the RMSE values it is evident that the interaction model indicates a slightly better with the RMSE having a value of 2.161 compared to 2.178. We can determine that the interaction model doesn't showcase that significant of a change in the model as the predictor variables as a whole are very significant in impacting the response variable of pm2.5.

## Poisson Regression Model

```
# Fit Poisson regression model using glmnet
set.seed(123)
cv.fit <- cv.glmnet(as.matrix(train[, c("CMAQ", "aod")]),
                    train$value, family = "poisson", nfolds = 5, alpha = 0)
# Identify optimal lambda value to properly run the training dataset
lambda <- cv.fit$lambda.min
# Generate model with optimal lambda value
poisson_model <- glmnet(as.matrix(train[, c("CMAQ", "aod")]), train$value,
                        family = "poisson", lambda = lambda)
# Test data predictions
test_predictions <- predict(poisson_model, newx = as.matrix(test[, c("CMAQ", "aod")]),
                            s = lambda)
# Obtaining root mean squared errors
rmse_2 <- sqrt(mean((test$value - test_predictions)^2))
rmse_2 #Store in rmse_2 for second data set analyzed
```

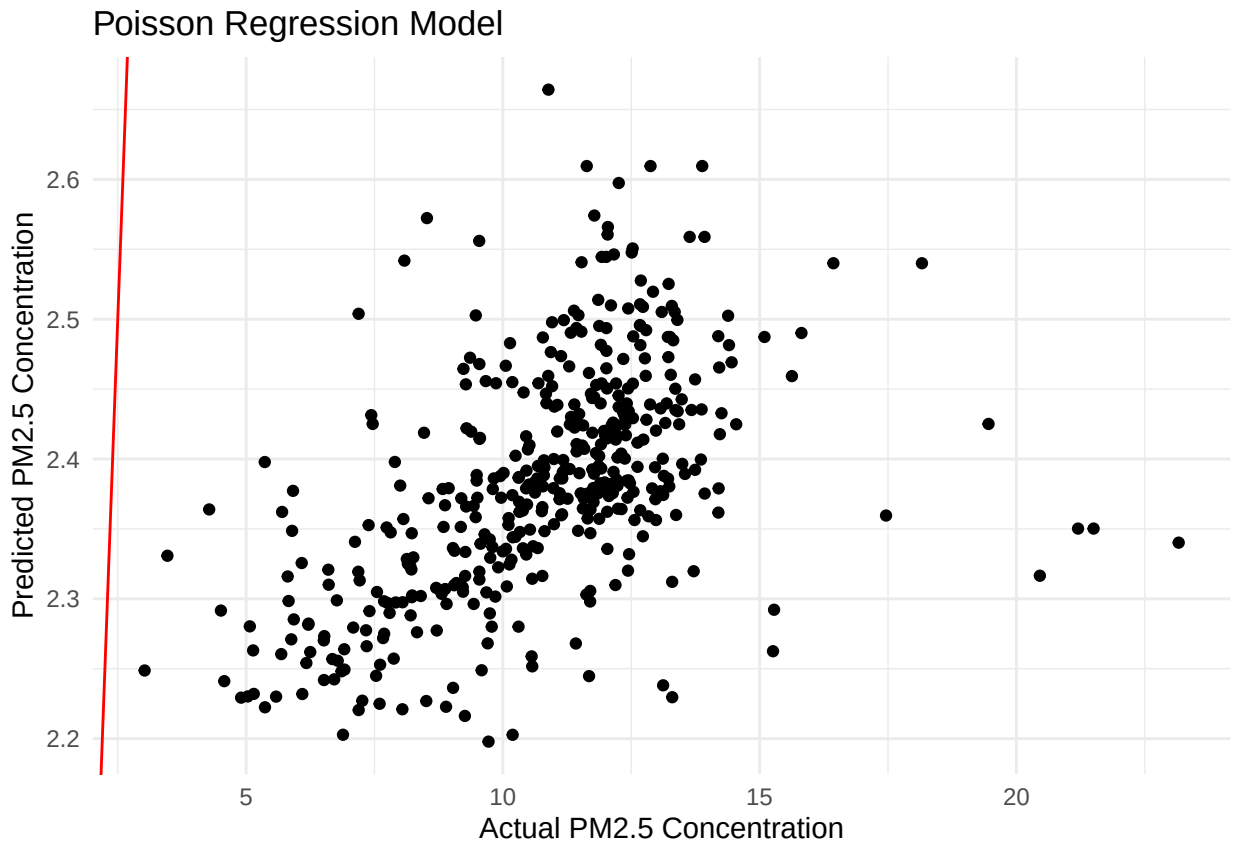
```
## [1] 8.779627
```

The poisson regression model is used to model count data, which in this case is for the count of the concentration pm2.5 in the atmosphere. While this variable is continuous, given the fact that it is slightly right skewed and non-negative makes it an appropriate test. There is a tuning parameter needed for the input, which controls the amount of regularization of the model. The glmnet package was installed to provide an optimal lambda value to conduct the test. The summary function simply provides the technical jargon within the possion model, however the rmse variable yields a value of 8.77 which is significantly higher than the linear model just done.

**Exploratory Analysis**

## Poisson Regression Prediction Performance Visualization

```
# Creating the scatterplot for the poisson model
ggplot(data = data.frame(value = test$value, test_predictions),
       aes(x = value, y = test_predictions)) +
geom_point() +
geom_abline(intercept = 0, slope = 1, color = "red") +
labs(title = "Poisson Regression Model",
x = "Actual PM2.5 Concentration",
y = "Predicted PM2.5 Concentration") +
theme_minimal()
```



We notice from the visualization here that the fit is not strong at all as the linear line is away from the cluster of the data. With that being said, we can conclude that the for the Poisson regression model to represent the data well other variables need to be included and that the predictor variables of CMAQ and aod are not sufficient enough to model pm2.5 accurately. Furthermore, in connecting back to the null dev stat (deviance) of the model, we can assume that this played a significant role in creating a large RMSE.

## Random Forest Regression Model

```
set.seed(123)
trainIndex <- createDataPartition(dat$value, p = 0.5, list = FALSE)
train <- dat[trainIndex, ] # Create training data set
test <- dat[-trainIndex, ] # Create test data set

# Train a random forest regression model based on ML binary trees
rf_fit <- randomForest(value ~ CMAQ + aod, data = train)
# Make predictions on the test data
predictions_rf <- predict(rf_fit, newdata = test)
rmse_3 <- sqrt(mean((test$value - predictions_rf)^2))
rmse_3 # Stores 3rd model results in rmse_3
```

```
## [1] 2.210527
```

```
summary(rf_fit)
```

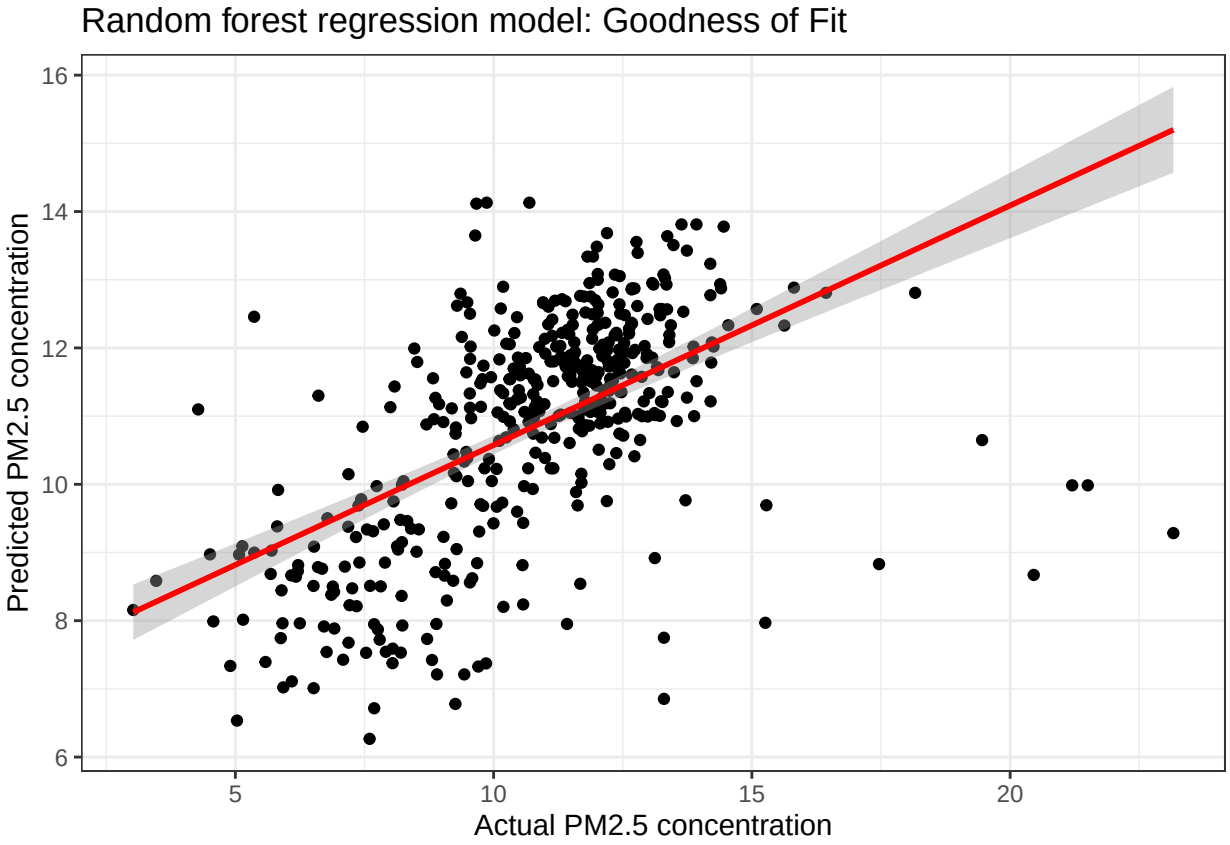
```
##                 Length Class  Mode
## call               3   -none- call
## type               1   -none- character
## predicted        440   -none- numeric
## mse              500   -none- numeric
## rsq              500   -none- numeric
## oob.times        440   -none- numeric
## importance         2   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            11   -none- list
## coefs              0   -none- NULL
## y                440   -none- numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

This model represents a random forest regresson model that is non parametric and does not make any assumptions about the data as it is purely based on random decision trees and averaging the respective outputs to make a decision using ML. The RMSE came out to be the lowest of any test at 2.21. Which could be in part due to the fact that this regression model handles both categorical and continous predictor variables as well as non linear relationships between the response and predictor variables. With air pollution data being pretty scattered and non-linear this model seems to be the best fit.

**Exploratory Analysis**

## Random Forest Regression Model Prediction Performance Visualization

```
ggplot(data = data.frame(actual = test$value, predicted = predictions_rf)) +
  geom_point(aes(x = actual, y = predicted)) +
  geom_smooth(aes(x = actual, y = predicted), method = "lm", color = "red") +
  labs(x = "Actual PM2.5 concentration", y = "Predicted PM2.5 concentration",
       title = "Random forest regression model: Goodness of Fit") +
  theme_bw()
```




A random forest regression model makes multiple decision trees based on random subsets of the continous response variable of the PM 2.5 level concentrations. Moreover each tree also calculates a random subset of both predictor variables: CMAQ and aod in attempting to predict the PM 2.5 concentration. The randomness of the test helps to prevent over fitting of the data and yields more accurate results, which was a major reason why this regression model was choosen. Consequently, the graph showcases a solid fit between the predicted and actual values. However there are some points that are off the regression line and would most likely yield higher residuals.

## K-Nearest Neighbors Classification-Regression Model

```r
# Train a k-nearest-neighbors model
knn_fit <- train(value ~ CMAQ + aod, data = train, method = "knn",
                 trControl = trainControl(method = "cv"))
# Make predictions on the test data
predictions_knn <- predict(knn_fit, newdata = test)
rmse_4 <- sqrt(mean((test$value - predictions_knn)^2))
rmse_4 #Stores 4th model results in variable rmse_4
```

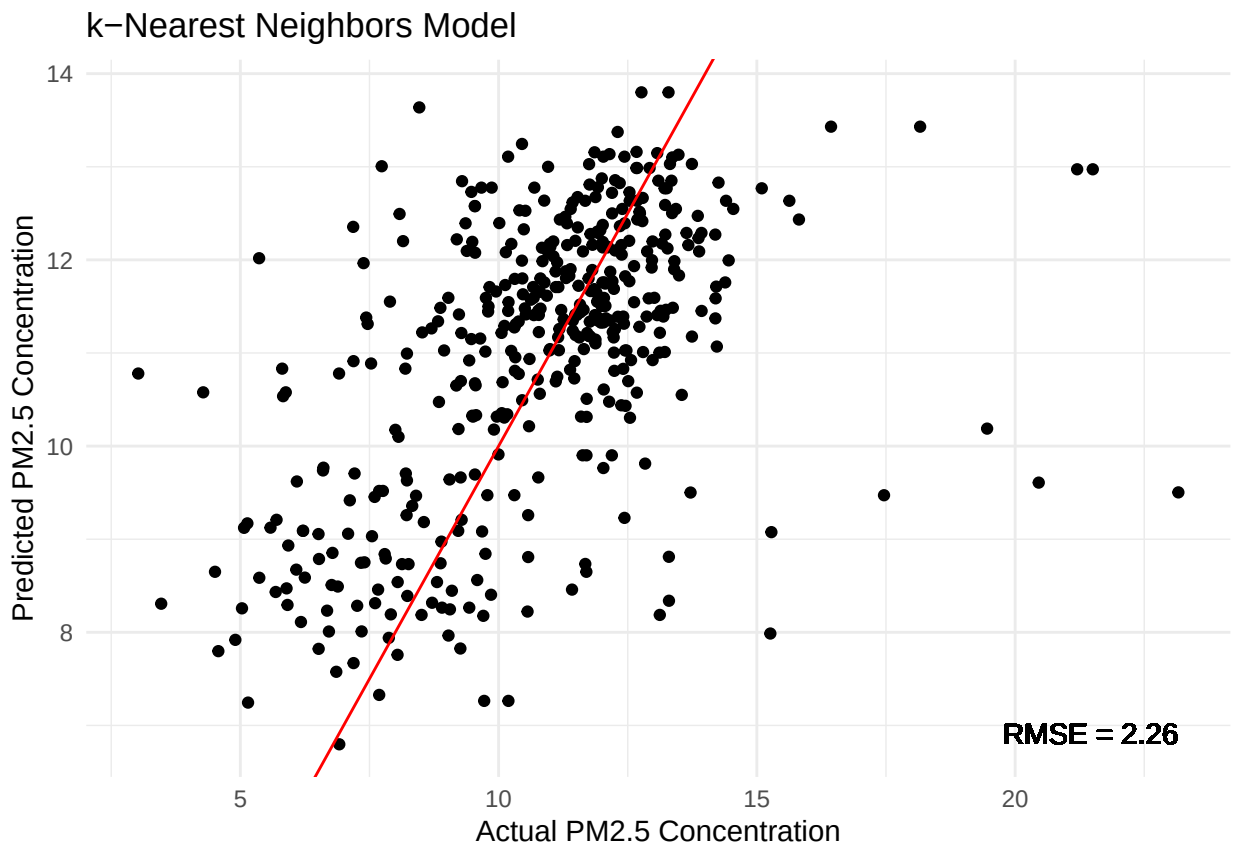
```
## [1] 2.257476
```

The K-nearest neighbors model works by finding k # of observations in the training dataset that are closest to an arbritary new observation that derives from the Euclidean distance and then taking a regression model of the average distance of the response variable. For the dat dataset the k was choosen to be the default value of 5 due to the fact that this yielded the most optimal rmse, therfore no additional optimal tuning parameters were needed.

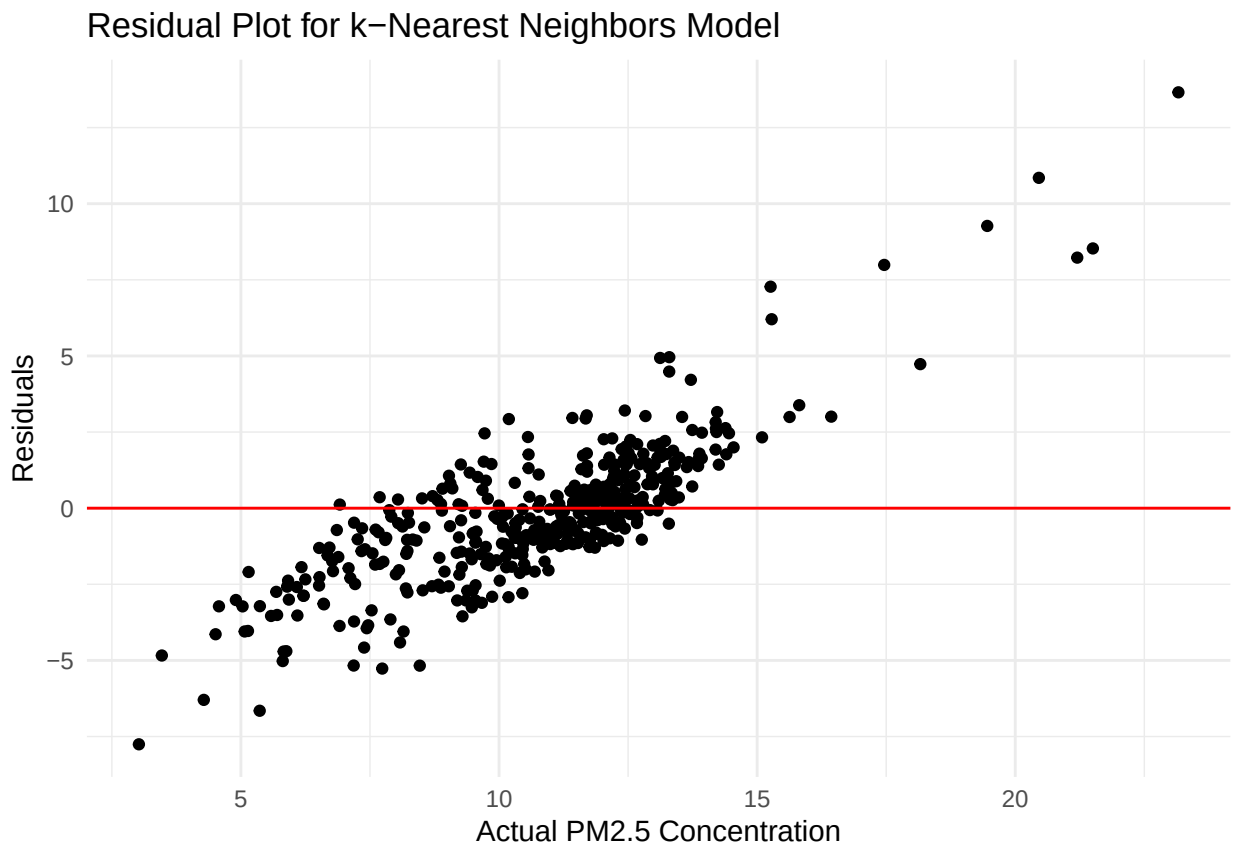
**Exploratory analysis**

## K-Nearest Neighbors Prediction Performance Visualization

```r
# Create a scatter plot of actual vs. predicted values
ggplot(data = data.frame(value = test$value, predictions_knn),
       aes(x = value, y = predictions_knn)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "k-Nearest Neighbors Model",
       x = "Actual PM2.5 Concentration",
       y = "Predicted PM2.5 Concentration") +
  theme_minimal() +
  geom_text(aes(x = max(test$value), y = min(predictions_knn),
                label = paste("RMSE =", round(rmse_4, 2))), hjust = 1, vjust = 0)
```




```
# Residual plot to analyze the RMSE value
residuals_knn <- test$value - predictions_knn
ggplot(data = data.frame(value = test$value, residuals_knn),
       aes(x = value, y = residuals_knn)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "red") +
  labs(title = "Residual Plot for k-Nearest Neighbors Model",
       x = "Actual PM2.5 Concentration",
       y = "Residuals") +
  theme_minimal()
```




The graph showcases a positive goodness of fit between the Actual PM2.5 and Predicted 2.5 Concentration values. Consequently, we can assume that the model is able to identify similar points for different k values based on the proximity to one another in the feature space of CMAQ and aod to predict pm2.5 levels for other data points. To test out the true effectiveness of the k-nearest Neighbhor model a residual plot was implemented. The plot showed a trend above and below the line y=0 which signifies that this model is not the best fit as there is evidence of non-linearity and unequal variance.

**Discussion**

## Model Performance Error Comparison

```
cat("Linear Regression RMSE:", rmse_int, "\n")
```

```
## Linear Regression RMSE: 2.246949
```

```
cat("Poisson Regression RMSE:", rmse_2, "\n")
```

```
## Poisson Regression RMSE: 8.779627
```

```
cat("Random Forest Regression RMSE:", rmse_3, "\n")
```

```
## Random Forest Regression RMSE: 2.210527
```

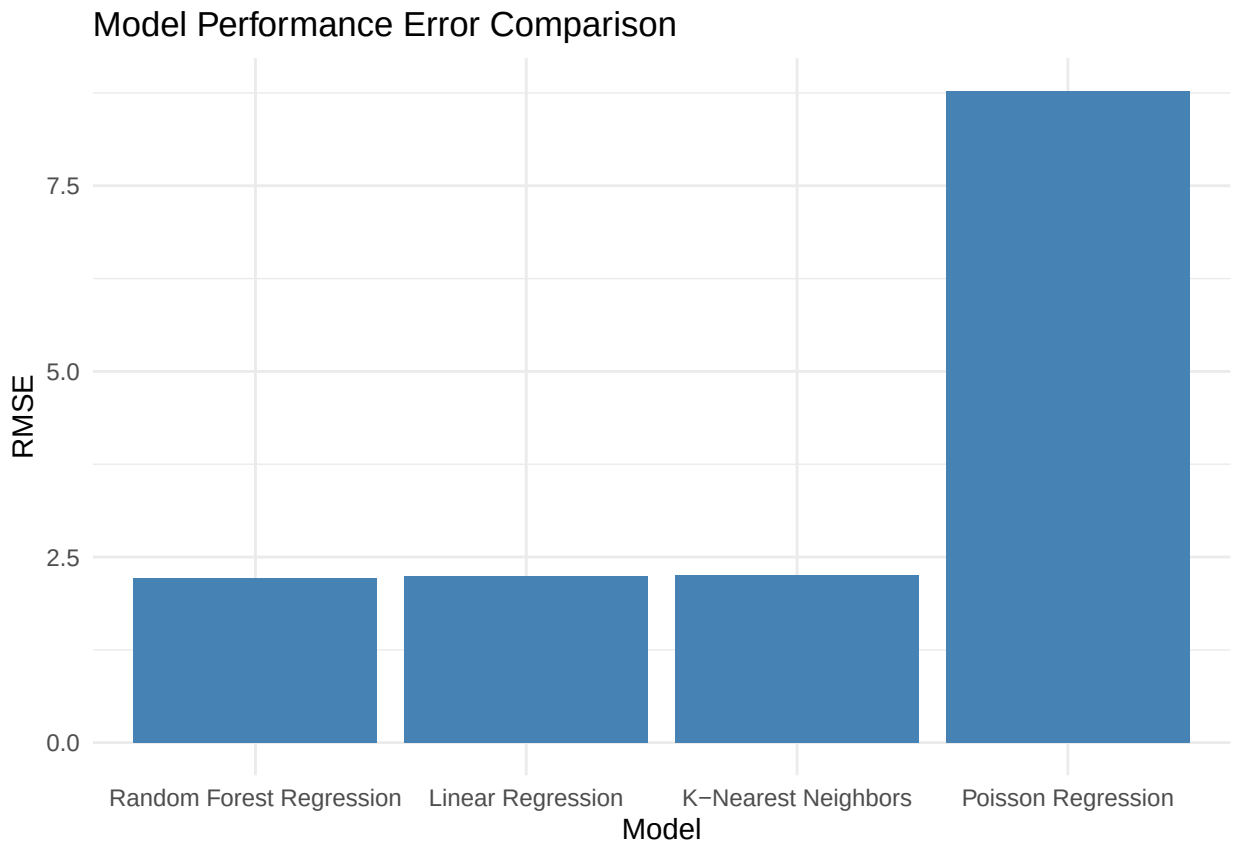
```
cat("K-Nearest Neighbors RMSE:", rmse_4, "\n")
```

```
## K-Nearest Neighbors RMSE: 2.257476
```

```
# Create a data frame of the RMSE values
rmse_df <- data.frame(
  model = c("Linear Regression", "Poisson Regression", "Random Forest Regression",
            "K-Nearest Neighbors"),
  rmse = c(rmse_int, rmse_2, rmse_3, rmse_4))

# Sort the data frame by rmse values
rmse_df <- rmse_df[order(rmse_df$rmse), ]

# Create a bar plot of the RMSE values
ggplot(rmse_df, aes(x = reorder(model, rmse), y = rmse)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(x = "Model", y = "RMSE", title = "Model Performance Error Comparison") +
  theme_minimal()
```




```
# Create a data frame with the RMSE values for each model
metrics_table <- data.frame(
  Model = c("Linear Regression", "Poisson Regression", "Random Forest Regression",
```

```
            "k-Nearest Neighbors"),
  RMSE = c(rmse_1, rmse_2, rmse_3, rmse_4),
  stringsAsFactors = FALSE
)

# Add a title to the table
names(metrics_table) <- c("Model", "RMSE")
title <- "Comparison of Prediction Metrics"
knitr::kable(metrics_table, col.names = c("Model", "RMSE"), caption = title)
```

Table 1: Comparison of Prediction Metrics

| Model | RMSE |
|---|---|
| Linear Regression | 2.279972 |
| Poisson Regression | 8.779627 |
| Random Forest Regression | 2.210527 |
| k-Nearest Neighbors | 2.257476 |

It is evident from the bar plot that the poisson regression has strikingly high root mean square error indicating a poor prediction performance. The linear regression and k-nearest neighbors models perform similar with almost identical RMSE values. We can see that the random forest regression has the lowest RMSE value compared to the linear regression, poisson regression, and k-nearest-neighbors regression model indicating a much better performance. Therefore, we will pick the Random Forest Regression model to use as the "best and final model".

*Question 1:* Based on the test set performance, let's now determine at what locations the model gives predictions that are closest and furthest from the observed values.

```
# Compute residuals and add them to the test data frame
test$residuals <- test$value - predictions_rf
#Prediction_rf signifies the random forest model being implemented

# Sort test data frame by absolute value of residuals
test <- test[order(abs(test$residuals)), ]
# Show the locations with the smallest absolute residuals
head(test)
```

```
## # A tibble: 6 x 52
##       id value  fips   lat    lon state county city   CMAQ  zcta zcta_~1 zcta_~2
##    <dbl> <dbl> <dbl> <dbl>  <dbl> <chr> <chr>  <chr> <dbl> <dbl>   <dbl>   <dbl>
## 1 30049.  7.53 30049  46.6 -112.  Mont~ Lewis~ Hele~  1.95 59636  5.05e6      12
## 2 18147. 11.8  18147  38.2  -87.0 Indi~ Spenc~ Dale  10.1  47523  1.70e8    3435
## 3  9009. 11.5   9009  41.3  -72.9 Conn~ New H~ New ~  8.60  6513  1.87e7   38978
## 4 13245. 13.5  13245  33.4  -82.0 Geor~ Richm~ Augu~ 10.9  30904  2.68e7   25666
## 5 28081. 11.8  28081  34.3  -88.8 Miss~ Lee    Tupe~  8.18 38801  1.96e8   29871
## 6 25013. 10.8  25013  42.1  -72.6 Mass~ Hampd~ Spri~  6.46  1103  1.19e6    2479
## # ... with 40 more variables: imp_a500 <dbl>, imp_a1000 <dbl>, imp_a5000 <dbl>,
## #   imp_a10000 <dbl>, imp_a15000 <dbl>, county_area <dbl>, county_pop <dbl>,
## #   log_dist_to_prisec <dbl>, log_pri_length_5000 <dbl>,
## #   log_pri_length_10000 <dbl>, log_pri_length_15000 <dbl>,
## #   log_pri_length_25000 <dbl>, log_prisec_length_500 <dbl>,
## #   log_prisec_length_1000 <dbl>, log_prisec_length_5000 <dbl>,
## #   log_prisec_length_10000 <dbl>, log_prisec_length_15000 <dbl>, ...
```

```r
# Sort test data frame by absolute value of residuals in descending order
test <- test[order(-abs(test$residuals)), ]
# Show the locations with the largest absolute residuals
head(test)
```

```
## # A tibble: 6 x 52
##      id value  fips   lat   lon state   county city   CMAQ  zcta zcta_~1 zcta_~2
##   <dbl> <dbl> <dbl> <dbl> <dbl> <chr>   <chr>  <chr> <dbl> <dbl>   <dbl>   <dbl>
## 1 6029.  23.2  6029  35.3 -119. Califo~ Kern   Bake~  5.75 93304  1.97e7   48731
## 2 6007.  20.5  6007  39.8 -122. Califo~ Butte  Not ~  5.36 95926  1.98e7   37725
## 3 6029.  21.5  6029  35.4 -119. Califo~ Kern   Bake~  5.75 93304  1.97e7   48731
## 4 6029.  21.2  6029  35.4 -119. Califo~ Kern   Bake~  5.75 93301  1.08e7   12248
## 5 4021.  19.5  4021  33.0 -112. Arizona Pinal  Mari~ 11.5  85138  1.92e8   33633
## 6 6073.  17.5  6073  33.1 -117. Califo~ San D~ Esco~  9.41 92025  5.71e7   49978
## # ... with 40 more variables: imp_a500 <dbl>, imp_a1000 <dbl>, imp_a5000 <dbl>,
## #   imp_a10000 <dbl>, imp_a15000 <dbl>, county_area <dbl>, county_pop <dbl>,
## #   log_dist_to_prisec <dbl>, log_pri_length_5000 <dbl>,
## #   log_pri_length_10000 <dbl>, log_pri_length_15000 <dbl>,
## #   log_pri_length_25000 <dbl>, log_prisec_length_500 <dbl>,
## #   log_prisec_length_1000 <dbl>, log_prisec_length_5000 <dbl>,
## #   log_prisec_length_10000 <dbl>, log_prisec_length_15000 <dbl>, ...
```

The locations with the smallest absolute residuals (i.e., the ones where the model's predictions are closest to the observed values) are shown in the first output. In this example, the top five locations with the smallest absolute residuals have `CMAQ` and `aod` values where the model is performing at the highest accuracy. The locations with the largest absolute residuals (i.e., the ones where the model's predictions are furthest from the observed values) are shown in the second output. In this example, the top five locations with the largest absolute residuals have `CMAQ` and `aod` values where the model is performing the poorest.

Based of our results, we can interpret that the relationships between the predictor variables and the response variable are more complex and cannot be fully captured by the linear model. With the random forest model using AI to generate random subsets of the predictor data from binary trees, it may have a better interpretation of a continuous data set compared to our other three model.

*Question 2:* Let's explore some variables that might affect the performance of the model. More specifically, we can examine certain regions of the country that the model performs better or worse. In the wrangling portion of our report, we split up the data set into multiple regions (East, West, North, and South) by filtering by the `lat` and `lon` variables. We believe these two variables greatly affect the performance of the model.

```r
# PM2.5 Monitors NORTH of the 38th Parallel (North Region)
set.seed(123)
trainIndex <- createDataPartition(north$value, p = 0.5, list = FALSE)
train <- north[trainIndex, ]
test <- north[-trainIndex, ]
rf_north <- randomForest(value ~ CMAQ + aod, data = train, importance = TRUE)
predictions_north <- predict(rf_north, newdata = test, type = "response")
rmse_north <- sqrt(mean((test$value - predictions_north)^2))
# PM2.5 Monitors SOUTH of the 38th Parallel (South Region)
set.seed(123)
trainIndex <- createDataPartition(south$value, p = 0.5, list = FALSE)
train <- south[trainIndex, ]
test <- south[-trainIndex, ]
rf_south <- randomForest(value ~ CMAQ + aod, data = train, importance = TRUE)
```

```r
predictions_south <- predict(rf_south, newdata = test, type = "response")
rmse_south <- sqrt(mean((test$value - predictions_south)^2))
# PM2.5 Monitors EAST of the -100th Meridian (East Region)
set.seed(123)
trainIndex <- createDataPartition(east$value, p = 0.5, list = FALSE)
train <- east[trainIndex, ]
test <- east[-trainIndex, ]
rf_east <- randomForest(value ~ CMAQ + aod, data = train, importance = TRUE)
predictions_east <- predict(rf_east, newdata = test, type = "response")
rmse_east <- sqrt(mean((test$value - predictions_east)^2))
# PM2.5 Monitors WEST of the -100th Meridian (West Region)
set.seed(123)
trainIndex <- createDataPartition(west$value, p = 0.5, list = FALSE)
train <- west[trainIndex, ]
test <- west[-trainIndex, ]
rf_west <- randomForest(value ~ CMAQ + aod, data = train, importance = TRUE)
predictions_west <- predict(rf_west, newdata = test, type = "response")
rmse_west <- sqrt(mean((test$value - predictions_west)^2))

# Data frame with the RMSE values for each region fitted by random forest model
metrics_table_2 <- data.frame(
  Model = c("North", "South", "East", "West"),
  RMSE = c(rmse_north, rmse_south, rmse_east, rmse_west),
  stringsAsFactors = FALSE)
names(metrics_table_2) <- c("Region", "RMSE")
title <- "Prediction Performance Comparison by Region"
knitr::kable(metrics_table_2, col.names = c("Region", "RMSE"), caption = title)
```

Table 2: Prediction Performance Comparison by Region

| Region | RMSE |
|---|---|
| North | 1.646194 |
| South | 2.308987 |
| East | 1.413656 |
| West | 3.200746 |

We can see that the model performs significantly better in the East region in comparison to the rest of the regions with a RMSE of just 1.41. This would make sense as the map that shows the ambient air pollution levels in the USA displays results in the east coast region that are consistent to this finding. The model performs the worst in the West out of the 4 regions as there appears to be the highest range (i.e. most variability in the PM2.5 values) in this region of the contiguous U.S. Additionally, the PM2.5 Monitors in the West appear to be more dispersed and farther away from each other compared to regions like the East and North that have much lower overall RMSE values. ## Model performance improvement Some variables that may improve the model's performance are state nominal GDP, county nominal GDP, and percentage of land mass in a state and county that are within a city limit. Nominal GDP is not only an important factor in measuring economic growth and status but most definitely a prime indicator of ambient air pollution concentration.

If we go back to our US map at the beginning which identifies the PM2.5 levels at each monitor, we notice a relatively consistent theme in states east of the Mississippi River and in California where the dots appear to be much lighter in color indicating much higher ambient air pollution levels. If we look at the top 25 states with the highest nominal GDPs as of 2021, California ranks in first place followed by Texas and a large pool of East coast states like New York, Pennsylvania, New Jersey, Ohio, Illinois, Indiana, and much more.

Percentage of a states' or countys' land mass that are within a city limit would be beneficial in improving the performance of our model because typically areas with higher concentrations of cities tend to have far greater pollution levels due to their higher populations, greater industrial presence, and large concentrations of automobiles. Additionally, weather-related variables like average annual precipitation in a county or zipcode or state can have a significant impact on the ambient air pollution levels in that area.

*Question 3:* Two candidates for replacing the use of ground-based monitors are numerical models like CMAQ and satellite-based observations such as AOD. Let's examine how well CMAQ and AOD predict ground-level concentrations of PM2.5. Given that the North region had the best overall performance with an RMSE of 0.385, we will use this dataframe including the set that included only the CMAQ and AOD variables (set 1). We will then see how the model performs without CMAQ and AOD. The second set will be a combination of all of the population variables in the North region dataset including the county and zipcode populations and their respective densities. The third set contains variables related to road length close to each PM2.5 monitor and the distance from the nearest road to a monitor.

```
# Split the data into training and testing sets
# use same sets for three different variables
set.seed(123)
trainIndex <- createDataPartition(north$value, p = 0.5, list = FALSE)
train <- north[trainIndex, ]
test <- north[-trainIndex, ]

# variable sets (set 2: population variables, set 3: road variables)
rf_pop <- randomForest(value ~ zcta_pop + county_pop + popdens_county +
                         popdens_zcta, data = train)
rf_road <- randomForest(value ~ log_dist_to_prisec + log_pri_length_5000 +
                          log_prisec_length_500, data = train)

# Make predictions on the test data & output respective RMSEs
predictions_pop <- predict(rf_pop, newdata = test, type = "response")
predictions_road <- predict(rf_road, newdata = test, type = "response")
rmse_pop <- sqrt(mean((test$value - predictions_pop)^2))
rmse_road <- sqrt(mean((test$value - predictions_road)^2))

# Create a data frame with the RMSE values for each model
metrics_table_3 <- data.frame(
  Model = c("CMAQ + AOD Model Variables", "Population Variables",
            "Road Length and Distance Variables"),
  RMSE = c(rmse_north, rmse_pop, rmse_road),
  stringsAsFactors = FALSE)
names(metrics_table_3) <- c("Model", "RMSE")
title <- "Variable Combinations Impact on PM2.5 Levels"
knitr::kable(metrics_table_3, col.names = c("Model", "RMSE"), caption = title)
```

Table 3: Variable Combinations Impact on PM2.5 Levels

| Model | RMSE |
|---|---|
| CMAQ + AOD Model Variables | 1.646194 |
| Population Variables | 1.863164 |
| Road Length and Distance Variables | 2.228982 |

Based on the RMSEs, the estimated values of air pollution from the Community Multiscale Air Quality computational model (CMAQ) and the Aerosol Optical Depth measurement from a NASA satellite are still

the most accurate in predicting ambient air pollution in comparison to population variables and road distance & length variables. A potential reason for population variables resulting in higher RMSEs is that some of these areas that have higher populations may have lower PM2.5 values relative to other areas up north. This can be due to a variety of factors like high precipitation levels and low urban development levels that can drastically impact the air quality. Vermont and New Hampshire are examples of such states.

Additionally, road variables like the count of primary and secondary road length in meters in a circle with a radius of 500 meters around the monitor and the distance to a primary or secondary road from the monitor can result in higher RMSEs because such variables don't take into account the amount of traffic on these roads and average volume of cars on these roads during rush hour times.

Lastly, from the table we can witness that the CMAQ + aod model has the lowest RMSE and hence performs the best in predicting ground level particulate concentration (pm2.5) compared to the other two predictor set models using different predictor variables from the original dataset.

*Question 4:* The dataset did not include Alaska and Hawaii. We believe our model would perform very well in those two states because both of these states have significantly low air quality indexes (AQI) relative to other states in the lower 48. Alaska and Hawaii are both significantly underdeveloped when it comes to urban growth and have unique climates that position them for remarkably low AQIs. Regardless of where the PM2.5 monitors would be placed within both of those states, the model would more than likely predict low ambient air pollution levels across the board.

**Reflection**

One of the most challenging parts of this process was to really contextualize what all of the results meant in the contexts of the 4 questions being asked. The first project let us choose our data set and research question and attack the tidying and wrangling part in our own manner, however with specific questions being asked about the model and its implications throughout different regions in America, both of us needed to step away from the code and really narrow our focus back into the objective of the project. Furthermore, we had initially included a logistic model that was classifying the data set as a classification model and comparing the PM 2.5 data compared the EPA regulated median value. Since, this was the goal of the last project we initially went off track and answered the wrong question. However, after speaking with the professor we were able to clarify the objective and added a new regression model to replace the logistic classification model.

The linear random forest and k nearest performed as well as expected with the random forest model performing extremely well with the lowest RMSE. As predicted the poisson did not compare as strongly to the others and a reason for that could include the fact that the Poisson model assumes that the mean and variance are equal, and this assumption is most probably not true for the P.M 2.5 concentration dataset. Given that fact, we believe more model research is necessary before implementation in order to understand if the model is appropriate for the structure of a specific data set.

Overall, this was a great learning experience and we have realized there is a lot more that goes into data science than what we had initially thought at the beginning of this semester. Moreover, within these models itself there is a lot more to unpack than the analysis we have conducted and we hope to build upon this project in our future professional work. For future work on this project, we would like to take into account other predictors not found in this data set like economic growth variables that would greatly improve the accuracy of our model in predicting the PM2.5 concentration nationwide.

**Work Distribution/Sources**

Vrishank (50%) Adhi (50%) Vrishank and Adhi both worked on the wrangling, model development, charts and visualizations, and explanations equally. https://www.stateofglobalair.org/ https://www.epa.gov/air-trends/particulate-matter-pm25-trends https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6864519/