Air Pollution Predictions

Name: Pranjal Adhikari

Introduction

The primary objective of this project is to explore multiple modeling approaches to predict the ambient air pollution concentrations across the United States. The models that will be utilized are linear regression and k-Nearest Neighbors (k-NN). Linear regression is a statistical model that determines the linear relationship between a dependent and independent variable. In single linear regression, there is only one independent variable, and in multiple linear regression there are multiple independent variables. k-NN is an algorithm that predicts the outcome of a data point based on the average value of the k-Nearest Neighbors. It measures the distances between multiple data points, and the result is the most prevalent value among the k-Nearest Neighbors.

The data used in this project includes the annual average concentrations of PM2.5 across EPA monitoring systems that are across the United Staets. The dataset can be found here. In this specific project, the predictor variables chosen from the dataset are `popdens_zcta`, `imp_a500`, and `nohs`. These predictors were chosen because I felt they were most important in determining the annual average PM2.5 concentration. The population, the impervious surface near the monitor, and the education of the county's citizens all have an impact on the air pollution of the United States.

```r
# load necessary libraries
library(tidyverse) # tidyverse
library(caret) # caret
library(randomForest)
library(plotROC) # ROC plot
```

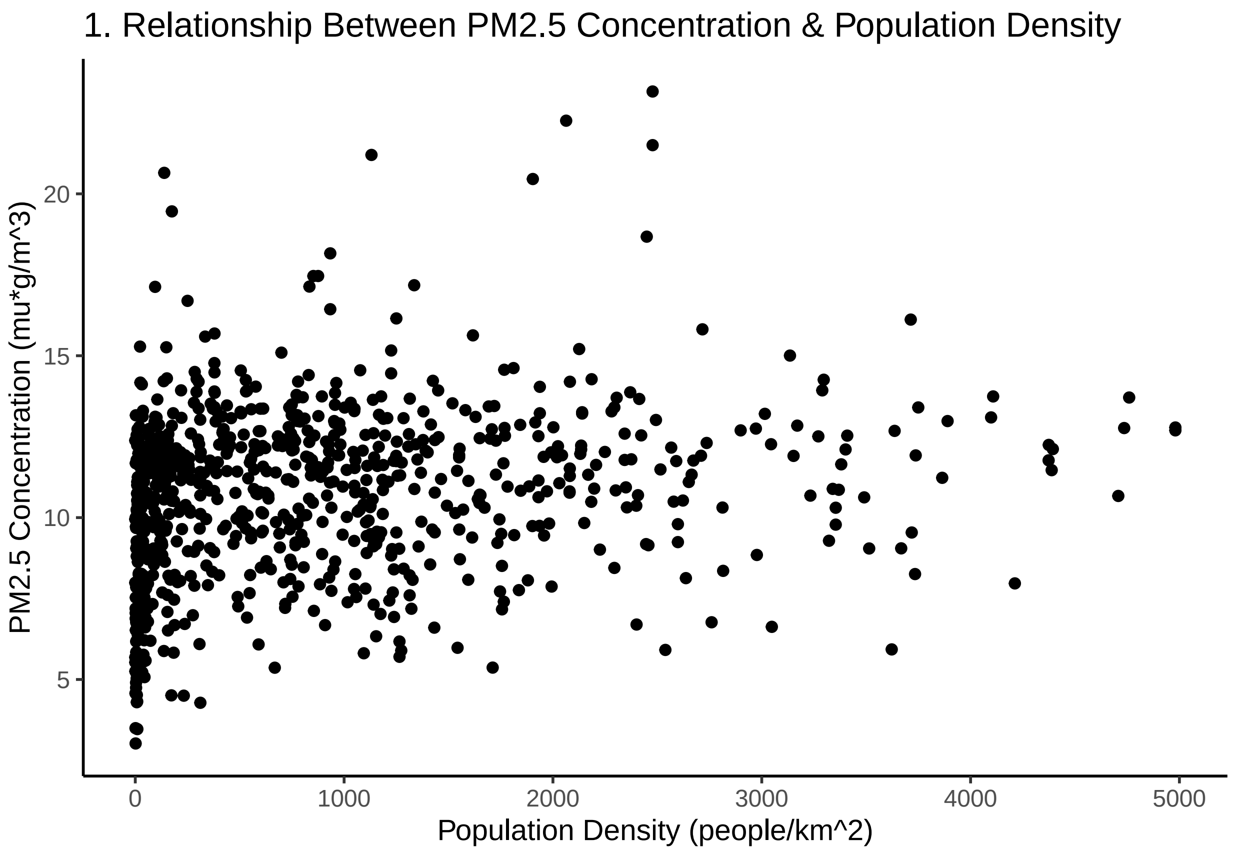
The packages/libraries used in this project include `tidyverse`, `caret`, and `plotROC`. The first allows for the usage of `tidyr` functions and `ggplot` for data visualization. The second package allows for the usage of the k-NN function within the project. Lastly, the third allows for the usage of `ggROC` to plot ROC curves of the prediction models.

```r
# read data into environment
emissions_dat <- read_csv("https://github.com/rdpeng/stat322E_public/raw/main/data/pm25_data.csv.gz")
```

To load the dataset that will be used in this project, it will first have to loaded into the environment. Since it is stored in a GitHub repository, the `read_csv` function is utilized then the data is saved into the environment using `<-` as the variable name `emissions_dat`.

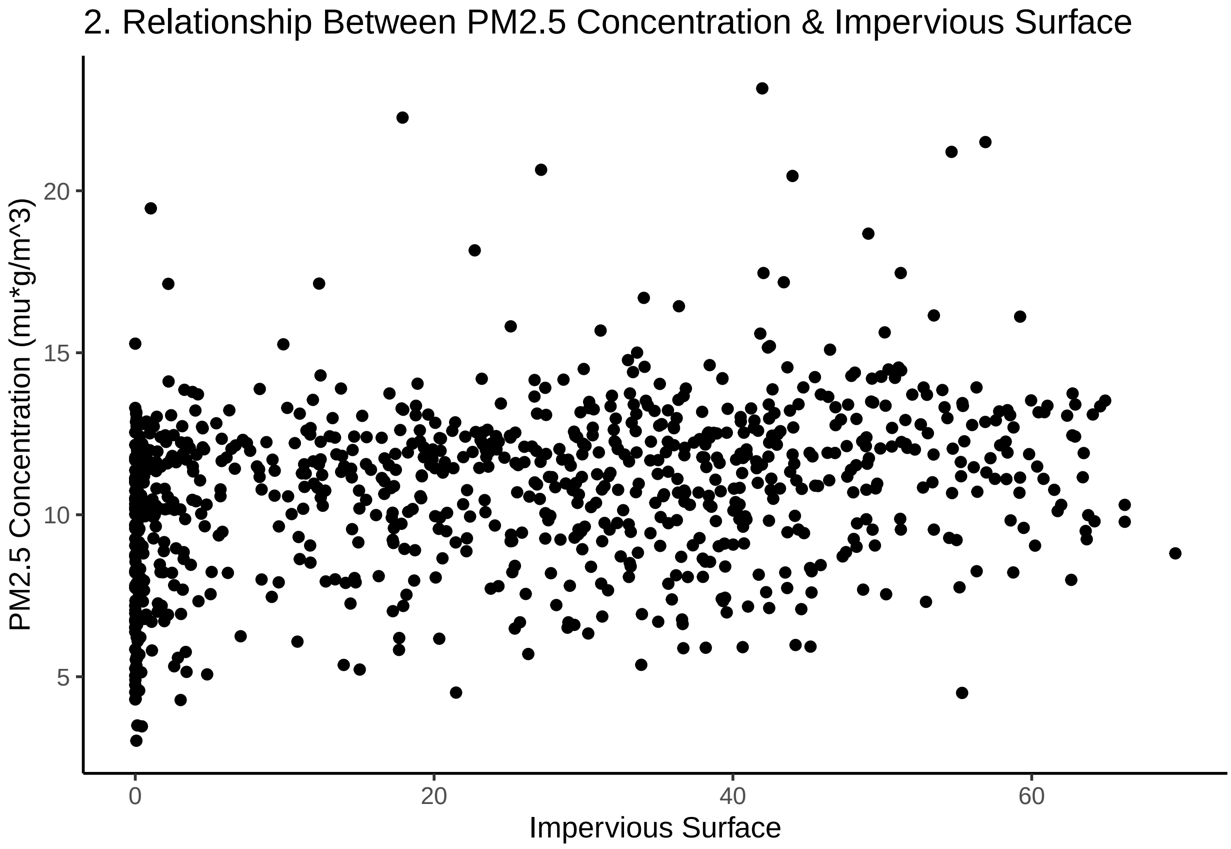
Prior to utilizing the models to predict the PM2.5 emissions, exploratory analysis is done to determine if there are any relationships between the predictors and the PM2.5 concentration for each monitor in the dataset. Below utilizes scatter plots to examine the relationship between the variables chosen in this project.

```r
# scatter plot of value and popdens_zcta
emissions_dat |> filter(popdens_zcta < 5000) |>
  ggplot(aes(y = value, x = popdens_zcta)) +
  geom_point() +
  labs(y = "PM2.5 Concentration (mu*g/m^3)", x = "Population Density (people/km^2)",
       title = "1. Relationship Between PM2.5 Concentration & Population Density") +
  theme_classic()
```



As seen in the plot above, there does not seem to be any direct relationship between the population density and PM2.5 concentration. The plot only encompasses population density values that is below 5000 people /km^2, as majority of the data points are below that value. Nonetheless, most of the values for the population density are concentrated below 500 people/km^2, and with the increase in population density values, the PM2.5 concentration do not seem to increase with it.

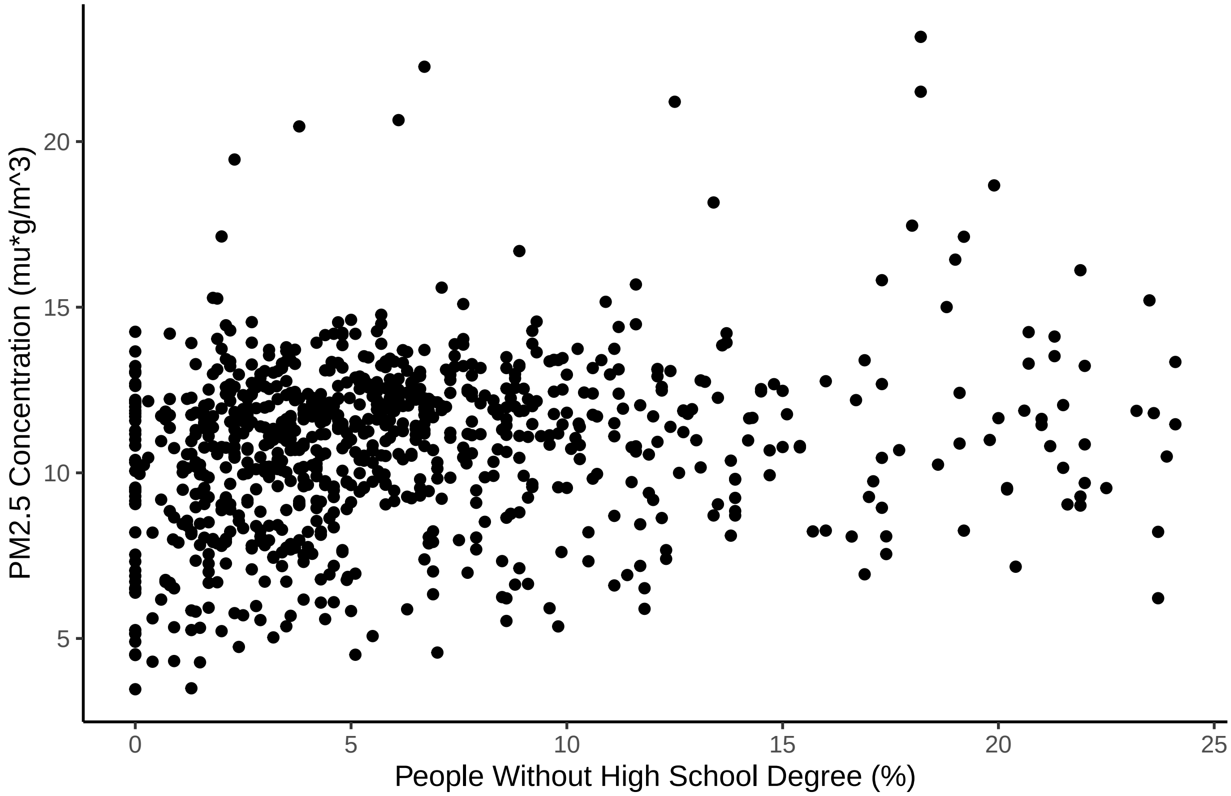
```
# scatter plot of value and imp_a500
emissions_dat |>
  ggplot(aes(y = value, x = imp_a500)) +
  geom_point() +
  labs(y = "PM2.5 Concentration (mu*g/m^3)", x = "Impervious Surface",
       title = "2. Relationship Between PM2.5 Concentration & Impervious Surface") +
  theme_classic()
```



The plot above depicts the relationship between the impervious surface and the PM2.5 concentration. As with the plot of the population density, there does not seem to be a relationship between the two variables in the above plot. With the increase of the impervious surface, the PM2.5 concentration does not increase, rather it seems to stay within the range of 5-15 μg/m^3.

```r
# scatter plot of value and nohs
emissions_dat |> filter(nohs < 25) |>
  ggplot(aes(y = value, x = nohs)) +
  geom_point() +
  labs(y = "PM2.5 Concentration (mu*g/m^3)",
       x = "People Without High School Degree (%)",
       title = "3. Relationship Between PM2.5 Concentration & High School Degree") +
  theme_classic()
```

3. Relationship Between PM2.5 Concentration & High School Degree



In this last scatter plot above, the relationship between the percentage of people without a high school degree and the PM2.5 concentration is shown. A large portion of the percentage is below 25%, thus the scatter plot has a maximum value of 25%. Consistent with what was seen in the last two scatter plots, there does not seem to be any relationship between the two variables. As the percentage of people without a high school degree increases, the PM2.5 concentration seems to be stagnant at around a level of 10 $\mu g/m^3$.

To determine how well the models predict the PM2.5 concentrations, the root mean-squared error (RSME) will be calculated for each of the two models. The model with the lowest RSME value will be chosen as the best and final to be utilized later with other predictor variables. For the final model that is chosen, my expectation for the RSME performance should be around a value of .3.

Wrangling

```
# mutate violation column if over PM2.5 limit
lin_emissions_dat_w_violation <- emissions_dat |>
        mutate(violation = ifelse(value > 12, 1, 0))

k_emissions_dat_w_violation <- emissions_dat |>
        mutate(violation = ifelse(value > 12, 1, 0))
```

In 2012, the EPA set a standard that restricted the annual average level of particle pollution (PM2.5) to be no more than 12 $\mu g/m^3$. The chunk above determines if the monitoring location is in violation of the national limit set by the EPA. The `mutate` function is used to create a new columns titled `violation` to show if the monitor is in violation of the PM2.5 concentration set by the EPA. For the `violation` column, the value 1 means the PM2.5 level is more than 12 $\mu g/m^3$, and the value 0 means it is not over the limit. This new `violation` column is created for all of the two models used in this project.

Results

The chunk below first saves the data into training and test dataset for each of the two models utilized in this project. This is done so that the prediction values that are calculated later will be unbiased without any impact by the other models.

```r
# save data into train & test variable for lin regression model
lin_train_data <- lin_emissions_dat_w_violation
lin_test_data <- lin_emissions_dat_w_violation

# save data into train & test variable for k-NN model
k_train_data <- k_emissions_dat_w_violation
k_test_data <- k_emissions_dat_w_violation
```

First, a linear regression model will be fit using `value` as the outcome, utilizing the three predictors chosen for this project: `popdens_zcta`, `imp_a500`, and `nohs`. It is then saved into the `lin_fit` variable for later use. Lastly, the `summary` function is utilized to check the residual standard error and the R-squared values to see how well the model performed on the dataset.

```r
# lin regression fit using chosen predictor variables
lin_fit <- lm(value ~ popdens_zcta + imp_a500 + nohs, data  = lin_train_data)

# summary of the fit
summary(lin_fit)
```

```
##
## Call:
## lm(formula = value ~ popdens_zcta + imp_a500 + nohs, data = lin_train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7122 -1.5829  0.3295  1.5793 11.6712
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.730e+00  1.473e-01  66.055  < 2e-16 ***
## popdens_zcta 1.914e-05  3.320e-05   0.577  0.56432
## imp_a500     3.295e-02  4.835e-03   6.814 1.77e-11 ***
## nohs         3.412e-02  1.195e-02   2.856  0.00439 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.473 on 872 degrees of freedom
## Multiple R-squared:  0.08628,    Adjusted R-squared:  0.08314
## F-statistic: 27.45 on 3 and 872 DF,  p-value: < 2.2e-16
```

Seen above, the residual stand error value was 2.473 and the R-squared value was .08314. It can be concluded that the model did not perform well on the dataset.

Next, the linear regression model is used to make predictions on the value of the PM2.5 concentrations. Utilizing the predictions made, it will be determined if the PM2.5 concentration is above the national limit set. Another column titled `predictions` and `predicted` will be made within the original training dataset to depict the prediction value, and whether the concentration value is above the limit. A value of 1 is added if the `predictions` is over .5, and 0 if below .5. Furthermore, the RMSE is calculated as well to determine the performance of the prediction.

```r
# predictions using lin regression fit
lin_predict <- lin_train_data |>
                select(violation) |>
                mutate(predictions = predict(lin_fit,  lin_train_data),
                predicted = ifelse(predictions > 12, 1, 0))

# calculate RMSE lin regression model
lin_RSME <- sqrt(mean((lin_train_data$violation - lin_predict$predicted)^2))
lin_RSME
```

```
## [1] 0.5753696
```

The value calculated above is the root mean-squared error (RSME) utilizing the linear regression model to predict the PM2.5 concentration, which resulted in a value of .575. The value itself is high, and it can be concluded that the linear regression model does not perform well to predict the PM2.5 concentration.

Next, the k-NN model is fit. The variable `value` is not used as an outcome, rather `violation` is used. But the same three predictors are utilized as seen before in the other model. Once again, the values are saved in a train and test dataset. However, the values are scaled because the k-NN algorithm is based on a point belonging to a specific class of 'k' nearest samples. Thus normalization is needed to solve the correct classification of the samples.

```r
# scale values for k-NN and save into train variable
k_train_data_scaled <- k_train_data |>
  select(-state, -county, -city) |>
  mutate(across(-violation, scale))

# scale values for k-NN and save into test variable
k_test_data_scaled <- k_test_data |>
  select(-state, -county, -city) |>
  mutate(across(-violation, scale))
```

After the values in the dataset are scaled, the k-NN model is fit, and the prediction whether the PM2.5 concentration is above or below the national limit is made. Once again, the `predictions` and `predicted` columns to show the prediction of the model, and assigns a value of 1 if the `predictions` is over .5, and 0 if below .5. The RSME is also calculated for the model.

```r
# k-NN model using the three predictors and violation as outcome
kNN_fit <- knn3(violation ~ popdens_zcta + imp_a500 + nohs,
                data = k_train_data_scaled, k = 5)

# determine whether the prediction is in violation of the national limit
kNN_predict <- k_train_data_scaled |>
                select(violation) |>
                mutate(predictions = predict(kNN_fit,  k_train_data_scaled)[,2],
                predicted = ifelse(predictions > 0.5, 1, 0))
```

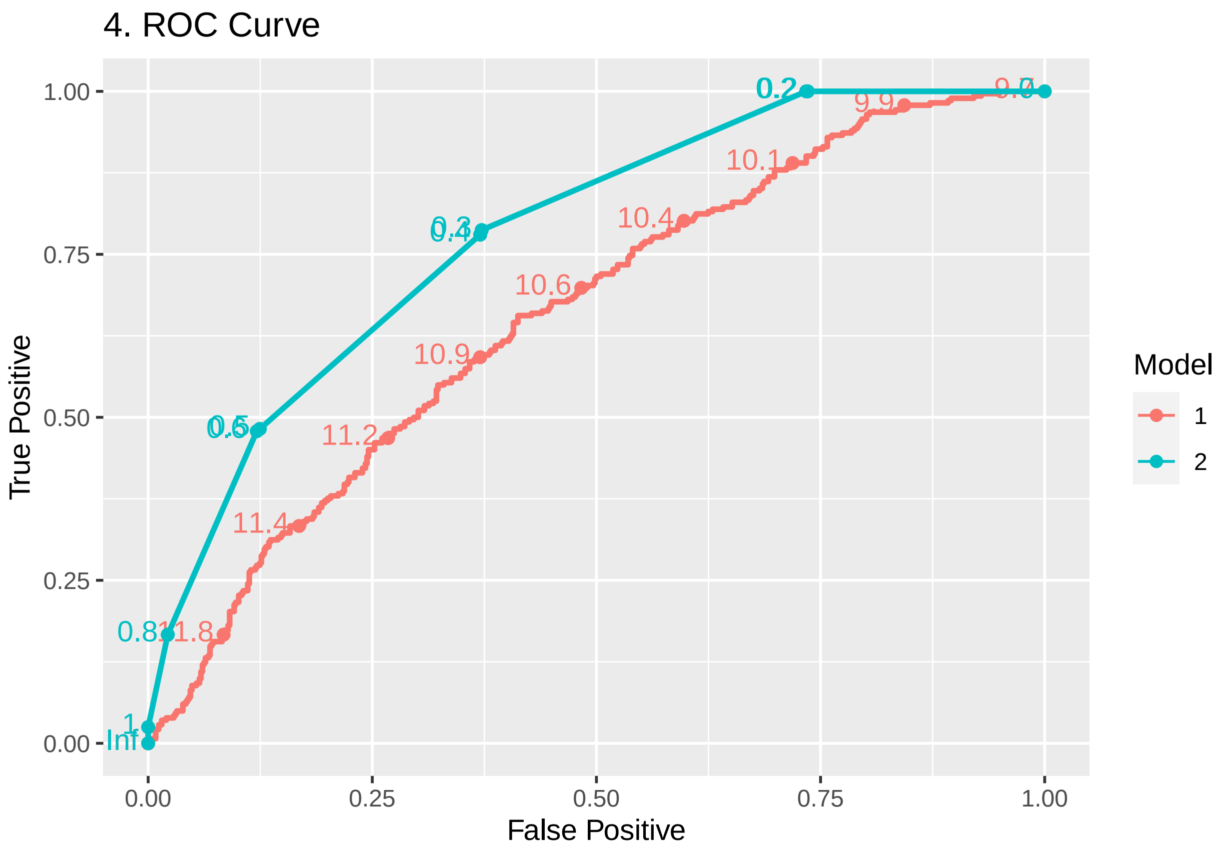
```r
# RSME calculated for k-NN
kNN_RSME <- sqrt(mean((k_train_data_scaled$violation - kNN_predict$predicted)^2))
kNN_RSME
```

```
## [1] 0.5
```

As seen above, the RSME value for the k-NN model is .5. This is comparable to the linear regression model's RSME, however it is the lowest out of the two.

For visualization, an ROC curve is created with both of the models to see the performance. The chunk below creates the ROC curves.

```r
# ROC curves of both models
lin_predict |>
  bind_rows(kNN_predict |> mutate(model = "k-NN"),
            .id = "model") |>
  ggplot(aes(d = violation,
             m = predictions,
             color = model)) +
  geom_roc() +
  scale_color_hue(labels = c("Linear", "k-NN")) |>
  labs(x = "False Positive",
       y = "True Positive",
       title = "4. ROC Curve",
       color = "Model")
```



As seen above, the k-NN model (Model 2) performed the best out of the two models in this project. The k-NN is closer to the upper left corner, meaning the model is better at ordering the data into the separate categories and the AUC is higher, meaning a larger portion of the plot is located underneath the curve.

Utilizing the best performing model, k-NN, the RSME will be evaluated on the test data set.

```
# RSME calculated for k-NN evaluated on the test data set
kNN_RSME <- sqrt(mean((k_test_data_scaled$violation - kNN_predict$predicted)^2))
kNN_RSME
```

```
## [1] 0.5
```

The RSME for the k-NN model evaluated on the test dataset is .5. This value is similar to the RSME calculated before for the k-NN model.

A table summarizing the prediction metrics of the two models used can be seen below.

```
# table with prediction metrics of the two models
data.frame(Models = c("Linear", "k-NN"),
           RSME = c(lin_RSME, kNN_RSME),
           Residual_standard_error = c(2.473, "-"),
           Num_neighbors_considered = c("-", "5"))
```

```
##   Models      RSME Residual_standard_error Num_neighbors_considered
## 1 Linear 0.5753696                   2.473                        -
## 2   k-NN 0.5000000                       -                        5
```

As described previously, the RSME of the k-NN is the lowest with .5, and the RSME of the linear regression model is the highest with .575.

Discussion

It can be concluded that best performing model in this project is the k-NN model. Calculating the RMSE of both models, the k-NN had the lowest value. Additionally, looking at the ROC curve the k-NN also performed better as the curve is closer towards having a AUC of 1.

```
# k-NN model using the three predictors and violation as outcome
kNN_fit <- knn3(violation ~ popdens_zcta + imp_a500 + nohs,
                data = k_train_data_scaled, k = 5)

# determine whether the prediction is in violation of the national limit
kNN_predict <- k_train_data_scaled |>
                select(violation) |>
                mutate(state = k_test_data$state) |>
                mutate(predictions = predict(kNN_fit,  k_train_data_scaled)[,2],
                predicted = ifelse(predictions > 0.5, 1, 0))

# number of correct predictions by state
kNN_predict |> group_by(state) |>
  mutate(ifequal = violation == predicted) |>
  filter(ifequal = TRUE) |> count() |>
  arrange(desc(n))
```

```
## # A tibble: 49 x 2
## # Groups:   state [49]
##    state              n
##    <chr>          <int>
##  1 California        85
##  2 Ohio              44
##  3 Illinois          38
##  4 Indiana           36
##  5 North Carolina    35
##  6 Pennsylvania      32
##  7 Michigan          30
##  8 Florida           29
##  9 Georgia           28
## 10 Texas             27
## # i 39 more rows
```

```r
# number of incorrect predictions by state
kNN_predict |> group_by(state) |>
  mutate(ifequal = violation == predicted) |>
  filter(ifequal != TRUE) |> count() |>
  arrange(desc(n))
```

```
## # A tibble: 43 x 2
## # Groups:   state [43]
##    state            n
##    <chr>        <int>
##  1 California      26
##  2 Ohio            17
##  3 Georgia         11
##  4 Indiana         11
##  5 Pennsylvania    11
##  6 Alabama         10
##  7 Illinois        10
##  8 Kentucky        10
##  9 Texas            8
## 10 New York         7
## # i 33 more rows
```

Based on the test performance, it seems that the k-NN model gives predictions that is closest to the observed value within the state of California. Furthermore, the model gives predictions that is furthest from the observed value within the state of California as well. An hypothesis as to why there is both good and bad performance in California could be that the predictors chosen for this project are not good predictors. They may not have as big of an impact on the PM2.5 concentration nor are they good indicators that have any effect on the concentration as originally thought.

With the results seen from the code chunk above, it seems that the eastern US region is where the model does the best and the worst. The states listed in the highest number of correct and incorrect predictions seem to be in the eastern US, except California. Some variables that are not included within this dataset that could improve the performance of the model would be the number of vehicles on the road. A large contributor of pollution would be from vehicles, especially in the US. Thus having one of the highest contributors of pollution included within the dataset could be a beneficial predictor to utilize to increase the performance of the model.

```
# k-NN model using the CMAQ & AOD with violation as outcome
kNN_fit <- knn3(violation ~ CMAQ + aod,
                data = k_test_data_scaled, k = 5)

# determine whether the prediction is in violation of the national limit
kNN_predict <- k_test_data_scaled |>
                select(violation) |>
                mutate(predictions = predict(kNN_fit,  k_test_data_scaled)[,2],
                predicted = ifelse(predictions > 0.5, 1, 0))

# RSME calculated for k-NN
sqrt(mean((k_test_data_scaled$violation - kNN_predict$predicted)^2))
```

```
## [1] 0.4482335
```

The chunk above utilizes the predictors `CMAQ` and `aod` to predict the violation of the PM2.5 concentration. With these predictors, the RSME is calculated to be .448. This value is lower than the RSME calculated utilizing `popdens_zcta`, `imp_a500`, and `nohs` as predictors. Thus, it can be concluded that prediction performance of the model is better when using `CMAQ` and `aod` than the original three predictors.

With the inclusion of Alaska and Hawaii into the dataset, I do not think the k-NN model will perform well in those two states. As seen above, the number of correct and incorrect predictions made by the model have states that are mostly in the eastern US. Since Alaska and Hawaii are in the western US, I do not think the model would be able to correctly predict the PM2.5 concentration for the two states.

Regarding the performance of the final prediction model, it did not perform as well as originally expected. I had expected the final model to have a RSME value of .3, while the best model, k-NN, actually had a value of .5. A reason to why the model did not perform well could be because the original three predictors chosen do not encompass the PM2.5 concentration as well as other variables. They may not have a large of an impact on the concentration as originally thought. However, the model's performance did improve after using the new `CMAQ` and `aod` candidates.

While working on the project, I utilized statology.org, stackoverflow.com, and geeksforgeeks.org for research to create the ROC curve specifically.