

Air Pollution Project

Ishita Jain

2024-08-20

Project: Air Pollution

Loading the data and packages needed.

```
library(tidyverse)
library(tidymodels)
tidymodels_prefer()

dat <- read_csv("pm25_data.csv.gz")

## Rows: 876 Columns: 50
## -- Column specification -----
## Delimiter: ","
## chr (3): state, county, city
## dbl (47): id, value, fips, lat, lon, CMAQ, zcta, zcta_area, zcta_pop, imp_a5...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

Title and Introduction

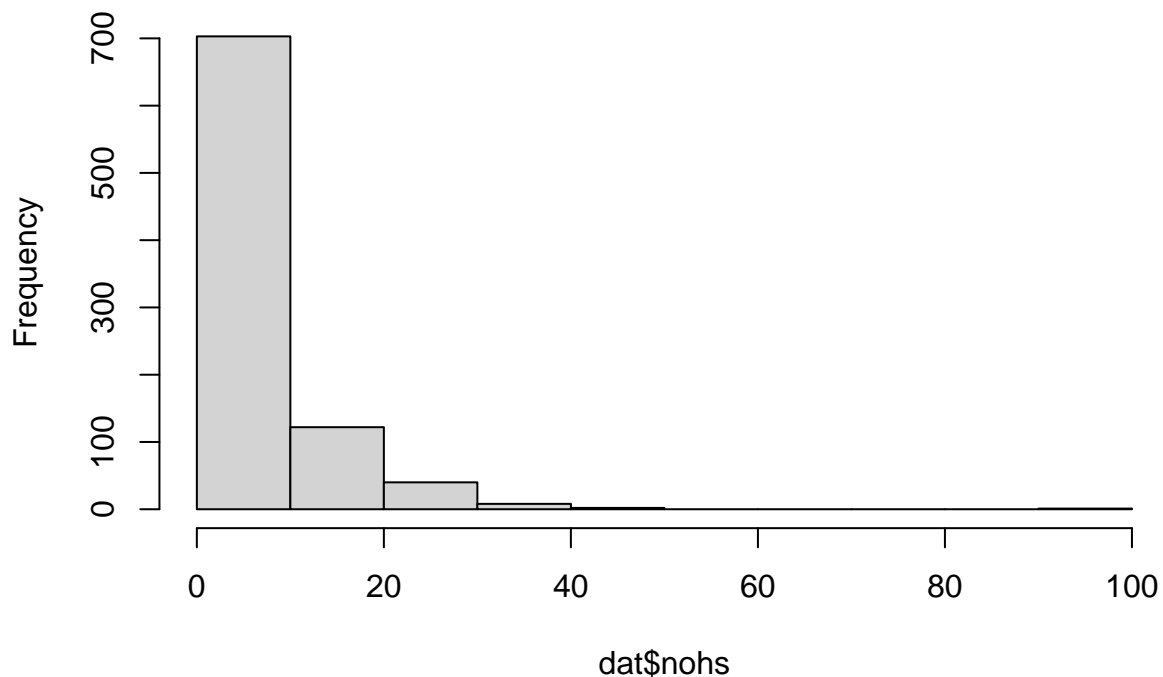
For this data set, I looked at three different modeling approaches including Linear Regression, K-Nearest Neighbors, and Random Forest. In terms of the predictors chosen I looked at the different variables presented and found ones I thought might help predict the average PM2.5 concentration at the monitors. I then looked at which variables would appear to have a significant linear relationship by running different summaries on different recipes and used those for the rest of the models. I also ran a histogram on the nohs variable and saw that it was not normal so I applied a logarithmic transformation which helped significantly. I expect the RMSE of the model to be between 2 and 5.

Wrangling

```
dat <- dat |>
  separate(col = id,
           into = c(NA, "monitor"))

hist(dat$nohs, main = "Histogram of People with No High School Degree")
```

Histogram of People with No High School Degree



```
dat <- dat |>
mutate(nohs = scale(nohs))
```

One of the columns in the data, id, appeared to hold extraneous information so I separated the data and kept the monitor number for later application. The data set appeared to be in a tidy format so not much else was needed.

Results

```
# Split into training and testing data sets
dat_split <- initial_split(dat)
dat_train <- training(dat_split)
dat_test <- testing(dat_split)
```

Linear Regression

```
rec <- dat_train |>
  recipe(value~CMAQ + aod + log_dist_to_prisec + log_nei_2008_pm25_sum_10000 + nohs) |>
  step_normalize(all_predictors()) |>
  step_scale(all_predictors())
```

```
lm_model <- linear_reg() |>
  set_engine("lm") |>
  set_mode("regression")

lm_wf <- workflow() |>
  add_recipe(rec) |>
  add_model(lm_model)

folds <- vfold_cv(dat_train, v = 10)
lm_res <- fit_resamples(lm_wf, resamples = folds, metrics = metric_set(rmse))
lm_metrics <- lm_res %>%
  collect_metrics() |>
  mutate(model = "linear regression")

lin_rmse = 2.1243151
```

K-Nearest Neighbors

```
knn_model <- nearest_neighbor(neighbors = tune("k")) %>%
  set_engine("knn") %>%
  set_mode("regression")

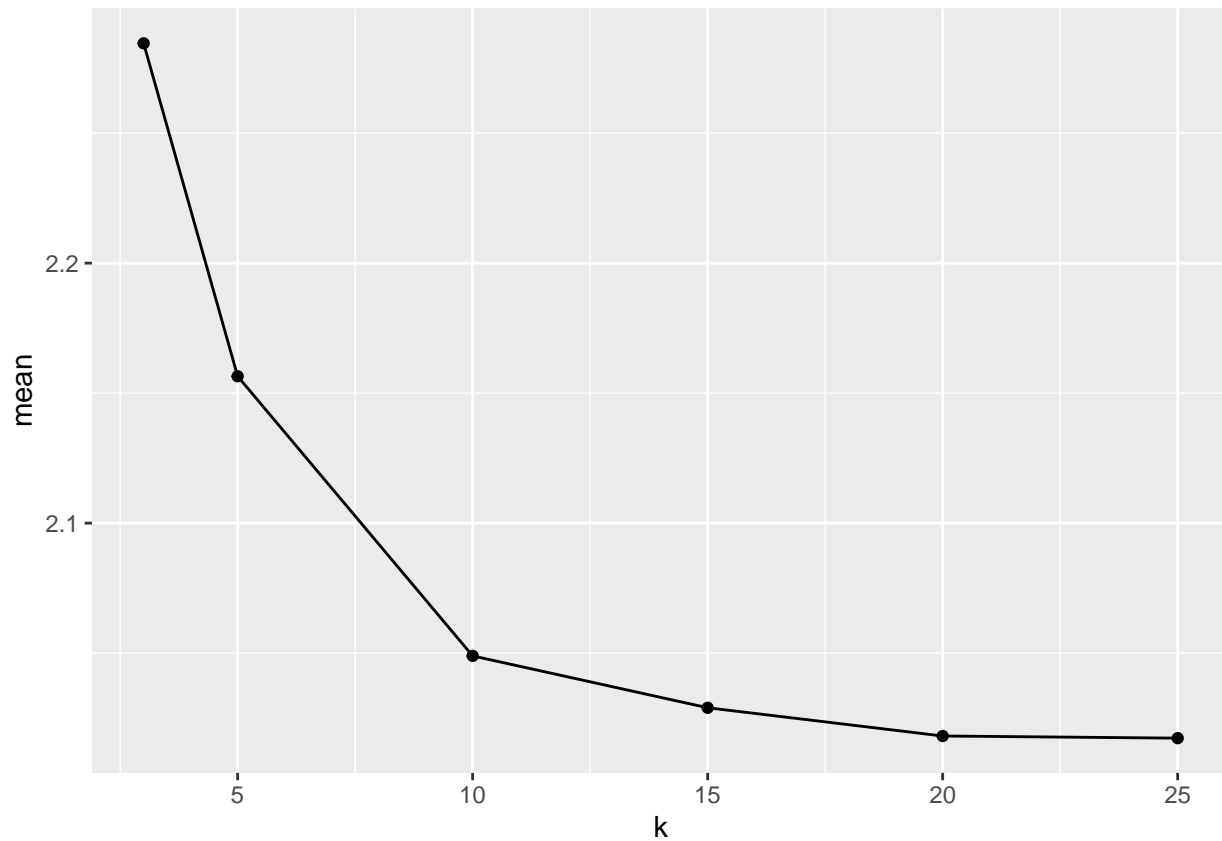
knn_wf <- workflow() %>%
  add_model(knn_model) %>%
  add_recipe(rec)

knn_res <- tune_grid(knn_wf, resamples = folds,
  grid = tibble(k = c(3, 5, 10, 15, 20, 25)),
  metrics = metric_set(rmse))

knn_res %>%
  collect_metrics()
```

```
## # A tibble: 6 x 7
##       k .metric .estimator mean      n std_err .config
##   <dbl> <chr>   <chr>    <dbl> <int>   <dbl> <chr>
## 1     3 rmse    standard  2.28    10  0.124 Preprocessor1_Model1
## 2     5 rmse    standard  2.16    10  0.107 Preprocessor1_Model2
## 3    10 rmse    standard  2.05    10  0.0923 Preprocessor1_Model3
## 4    15 rmse    standard  2.03    10  0.0931 Preprocessor1_Model4
## 5    20 rmse    standard  2.02    10  0.0969 Preprocessor1_Model5
## 6    25 rmse    standard  2.02    10  0.0986 Preprocessor1_Model6
```

```
knn_res %>%
  collect_metrics() %>%
  filter(.metric == "rmse") %>%
  ggplot(aes(k, mean)) +
  geom_point() +
  geom_line()
```



#Thus k neighbors of 20 seem to have the best model since it has the lowest rmse

```
knn_model <- nearest_neighbor(neighbors = 20) %>%
  set_engine("kkn") %>%
  set_mode("regression")

knn_wf <- workflow() %>%
  add_model(knn_model) %>%
  add_recipe(rec)

knn_res <- fit_resamples(knn_wf, resamples = folds, metrics = metric_set(rmse))
k_metrics <- knn_res %>%
  collect_metrics() %>%
  filter(.metric == "rmse") %>%
  mutate(model = "k_NN")
k_20_rmse = 2.008735
```

Random Forest Model

```
rf_model <- rand_forest(mtry = tune("mtry"),
  min_n = tune("min_n")) %>%
  set_engine("ranger") %>%
  set_mode("regression")
```

```
rf_wf <- workflow() %>%
  add_recipe(rec) %>%
  add_model(rf_model)

## Fit model over grid of tuning parameters
rf_res <- tune_grid(rf_wf, resamples = folds,
  grid = expand_grid(mtry = c(1, 2, 5),
    min_n = c(3, 5)))

rf_res %>%
  collect_metrics()
```

```
## # A tibble: 12 x 8
##   mtry min_n .metric .estimator mean n std_err .config
##   <dbl> <dbl> <chr>   <chr>   <dbl> <int> <dbl> <chr>
## 1     1     3 rmse    standard 1.92  10  0.0893 Preprocessor1_Model11
## 2     1     3 rsq     standard 0.455 10  0.0413 Preprocessor1_Model11
## 3     2     3 rmse    standard 1.91  10  0.0896 Preprocessor1_Model12
## 4     2     3 rsq     standard 0.454 10  0.0416 Preprocessor1_Model12
## 5     5     3 rmse    standard 1.96  10  0.0984 Preprocessor1_Model13
## 6     5     3 rsq     standard 0.429 10  0.0440 Preprocessor1_Model13
## 7     1     5 rmse    standard 1.92  10  0.0892 Preprocessor1_Model14
## 8     1     5 rsq     standard 0.453 10  0.0415 Preprocessor1_Model14
## 9     2     5 rmse    standard 1.92  10  0.0902 Preprocessor1_Model15
## 10    2     5 rsq     standard 0.452 10  0.0411 Preprocessor1_Model15
## 11    5     5 rmse    standard 1.97  10  0.0995 Preprocessor1_Model16
## 12    5     5 rsq     standard 0.426 10  0.0440 Preprocessor1_Model16
```

```
rf_res %>%
  show_best(metric = "rmse")
```

```
## # A tibble: 5 x 8
##   mtry min_n .metric .estimator mean n std_err .config
##   <dbl> <dbl> <chr>   <chr>   <dbl> <int> <dbl> <chr>
## 1     2     3 rmse    standard 1.91  10  0.0896 Preprocessor1_Model12
## 2     2     5 rmse    standard 1.92  10  0.0902 Preprocessor1_Model15
## 3     1     3 rmse    standard 1.92  10  0.0893 Preprocessor1_Model11
## 4     1     5 rmse    standard 1.92  10  0.0892 Preprocessor1_Model14
## 5     5     3 rmse    standard 1.96  10  0.0984 Preprocessor1_Model13
```

```
rf_metrics <- rf_res |>
  collect_metrics() |>
  filter(.metric == "rmse") |>
  filter(mtry == 2 & min_n == 3) |>
  mutate(model = "random forest") |>
  arrange(mean)

randFor_rmse = 1.854144
```

Summarizing Results

```
combined <- bind_rows(lm_metrics, k_metrics, rf_metrics)
```

```
combined |>  
  group_by(model) |>  
  summarize(mean) |>  
  rename("RMSE" = mean) |>  
  arrange(RMSE)
```

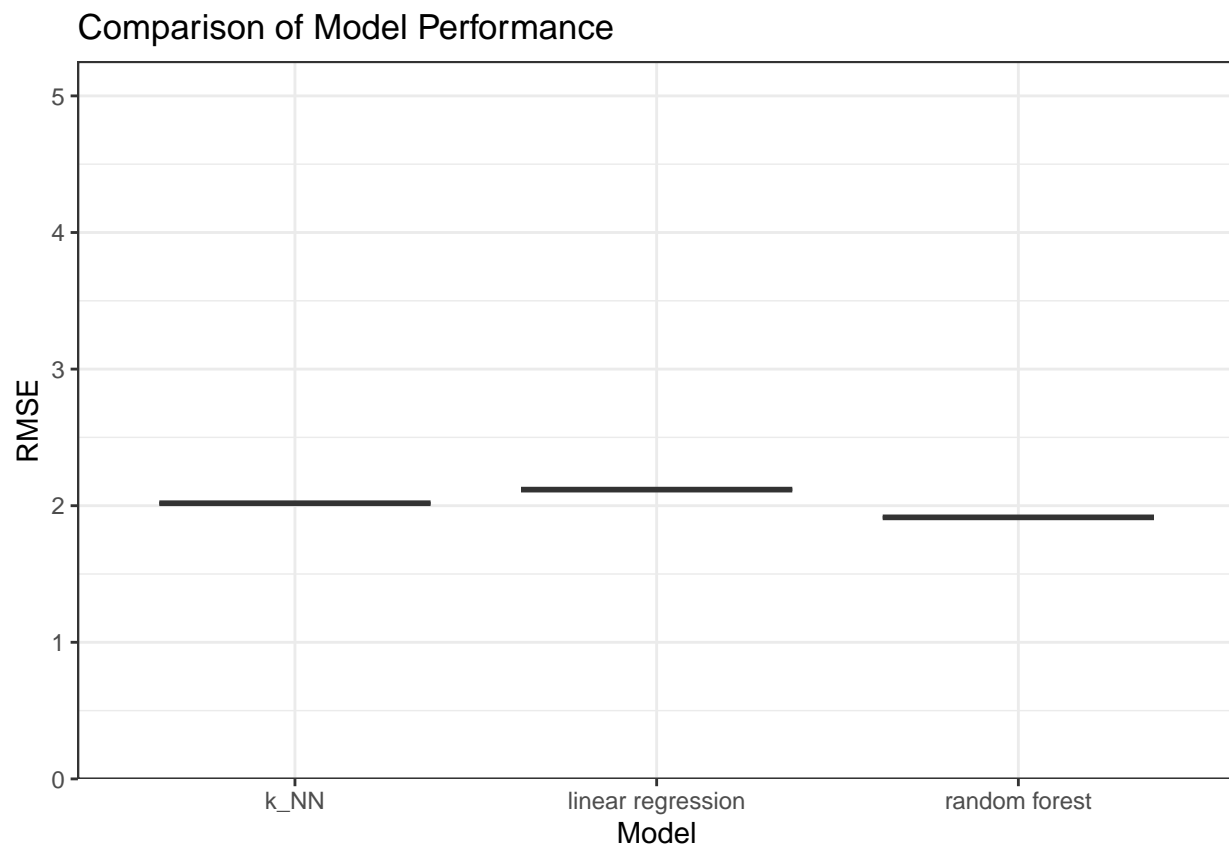
```
## # A tibble: 3 x 2  
##   model      RMSE  
##   <chr>    <dbl>  
## 1 random forest  1.91  
## 2 k_NN          2.02  
## 3 linear regression  2.12
```

```
# Gather RMSE values into a data frame
```

```
rmse_df <- bind_rows(lm_res, knn_res, rf_res)
```

```
# Create a plot of the RMSE values for each model
```

```
ggplot(combined, aes(x = model, y = mean)) +  
  geom_boxplot() +  
  scale_y_continuous(limits = c(0, 5), expand = expansion(mult = c(0, 0.05))) +  
  labs(x = "Model", y = "RMSE", title = "Comparison of Model Performance") +  
  theme_bw()
```



```

rf_model <- rand_forest(mtry = 5, min_n = 5) %>%
  set_engine("ranger") %>%
  set_mode("regression")

rf_wf <- workflow() %>%
  add_recipe(rec) %>%
  add_model(rf_model)

rf_fit <- rf_wf %>%
  fit(dat_train)

rf_pred <- predict(rf_fit, dat_test) %>%
  bind_cols(dat_test)

dat_test$diffrences <- rf_pred$.pred - dat_test$value

rf_rmse <- sqrt(mean(dat_test$diffrences^2, na.rm = TRUE))
rf_rmse

```

```
## [1] 2.009844
```

Primarily I split the data into training and testing data sets with a 75-25 split. I created 2 different models all using the same recipe to test which approach works the best. They included linear regression, k-nearest neighbors and random forest. The model with the lowest RMSE appears to be the Random Forest model showing it is the best approach.

Discussion

```

dat_test |>
  group_by(state) |>
  summarize(avg = mean(diffrences), n = n()) |>
  arrange(abs(avg))

```

```

## # A tibble: 42 x 3
##   state      avg      n
##   <chr>    <dbl> <int>
## 1 Arkansas -0.0173     3
## 2 Oregon   -0.0307     4
## 3 Tennessee -0.0372     1
## 4 Indiana    0.0588     9
## 5 Alabama   -0.0685     6
## 6 Michigan  -0.0786    14
## 7 Illinois  -0.107     10
## 8 Iowa       0.127      5
## 9 North Dakota -0.144     1
## 10 Virginia  0.200      7
## # i 32 more rows

```

1. For the most part, the locations that are closest from observed values appear to be near a water source and opposite for the far ones. This might be due to the clarity of the air when it is near the coast.

2. I do not think there are exact regions for the model that I have where they perform better or worse since they appear mainly scattered except for the coasts. I think information about the locations car pollution might be a strong predictor that could have helped.
3. The model seems to be weaker when CMAQ and AOD are not included in the model with higher residuals.
4. I do not think the model will perform quite well for Hawaii or Alaska since these states were left out of the data and the weather conditions are drastically different there compared to other US locations. It was very challenging for me to come up with a visualization for the rmse but I have learned to keep trying and experimenting to reach a conclusion. It performed as I expected but on the lower spectrum of what I thought for my models. I did this project on my own.