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")
```

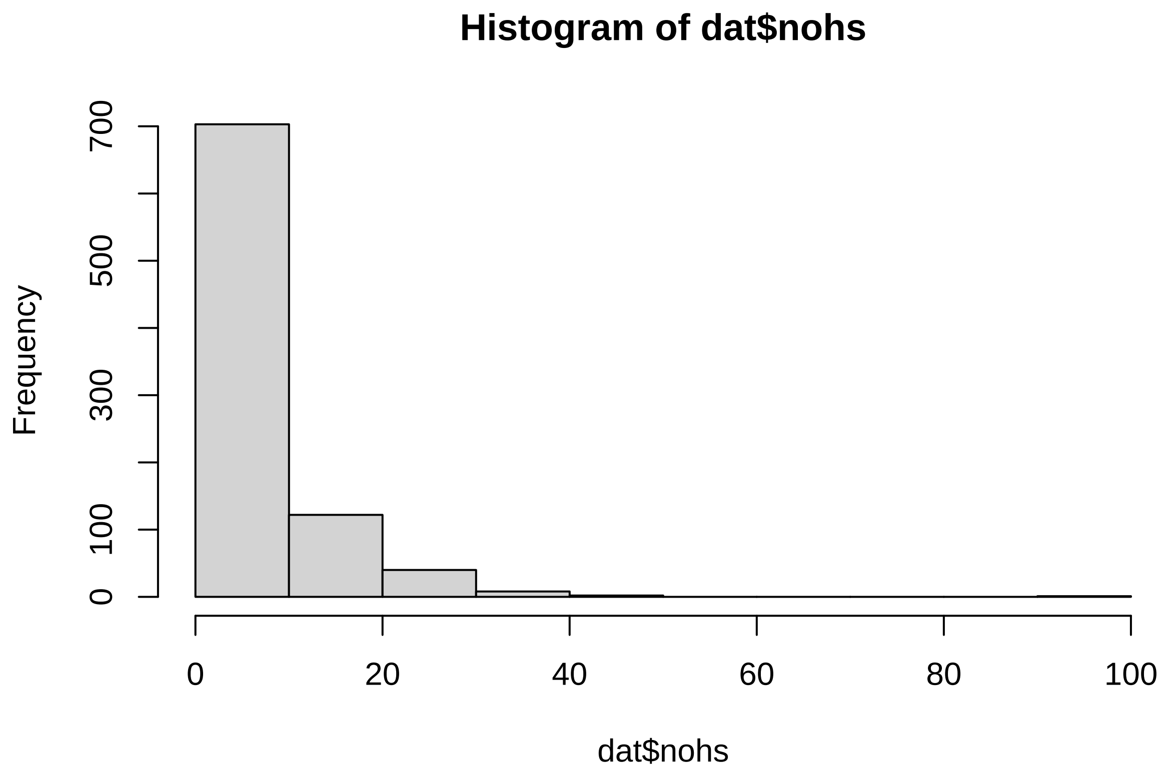
```
## Rows: 876 Columns: 50
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): state, county, city
## dbl (47): id, value, fips, lat, lon, CMAQ, zcta, zcta_area, zcta_pop, imp_a5...
##
## i Use `spec()` to retrieve the full column specification 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)
```




```
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())
```

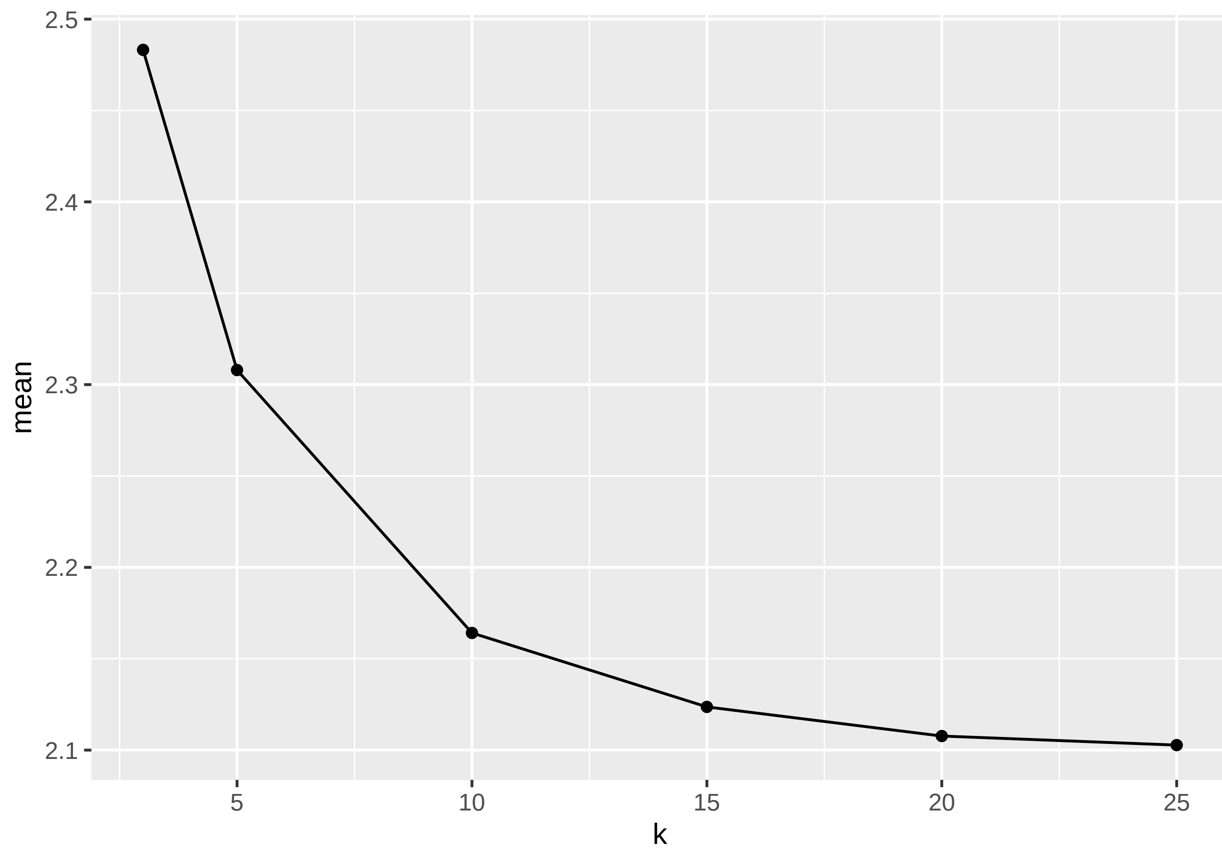
```
lm_model <- linear_reg() |>
  set_engine("lm") |>
  set_mode("regression")

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("kknn") %>%
  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    2.48    10   0.131 Preprocessor1_Model1
## 2     5 rmse    standard    2.31    10   0.121 Preprocessor1_Model2
## 3    10 rmse    standard    2.16    10   0.107 Preprocessor1_Model3
## 4    15 rmse    standard    2.12    10   0.105 Preprocessor1_Model4
## 5    20 rmse    standard    2.11    10   0.106 Preprocessor1_Model5
## 6    25 rmse    standard    2.10    10   0.105 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("kknn") %>%
  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")
```

```r
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   2.02     10  0.0991 Preprocessor1_Model1
##  2     1     3 rsq     standard   0.375    10  0.0251 Preprocessor1_Model1
##  3     2     3 rmse    standard   2.01     10  0.0845 Preprocessor1_Model2
##  4     2     3 rsq     standard   0.377    10  0.0229 Preprocessor1_Model2
##  5     5     3 rmse    standard   2.06     10  0.0837 Preprocessor1_Model3
##  6     5     3 rsq     standard   0.347    10  0.0247 Preprocessor1_Model3
##  7     1     5 rmse    standard   2.02     10  0.0979 Preprocessor1_Model4
##  8     1     5 rsq     standard   0.375    10  0.0257 Preprocessor1_Model4
##  9     2     5 rmse    standard   2.00     10  0.0847 Preprocessor1_Model5
## 10     2     5 rsq     standard   0.382    10  0.0231 Preprocessor1_Model5
## 11     5     5 rmse    standard   2.05     10  0.0831 Preprocessor1_Model6
## 12     5     5 rsq     standard   0.352    10  0.0240 Preprocessor1_Model6
```

```r
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     5 rmse    standard    2.00    10  0.0847 Preprocessor1_Model5
## 2     2     3 rmse    standard    2.01    10  0.0845 Preprocessor1_Model2
## 3     1     3 rmse    standard    2.02    10  0.0991 Preprocessor1_Model1
## 4     1     5 rmse    standard    2.02    10  0.0979 Preprocessor1_Model4
## 5     5     5 rmse    standard    2.05    10  0.0831 Preprocessor1_Model6
```

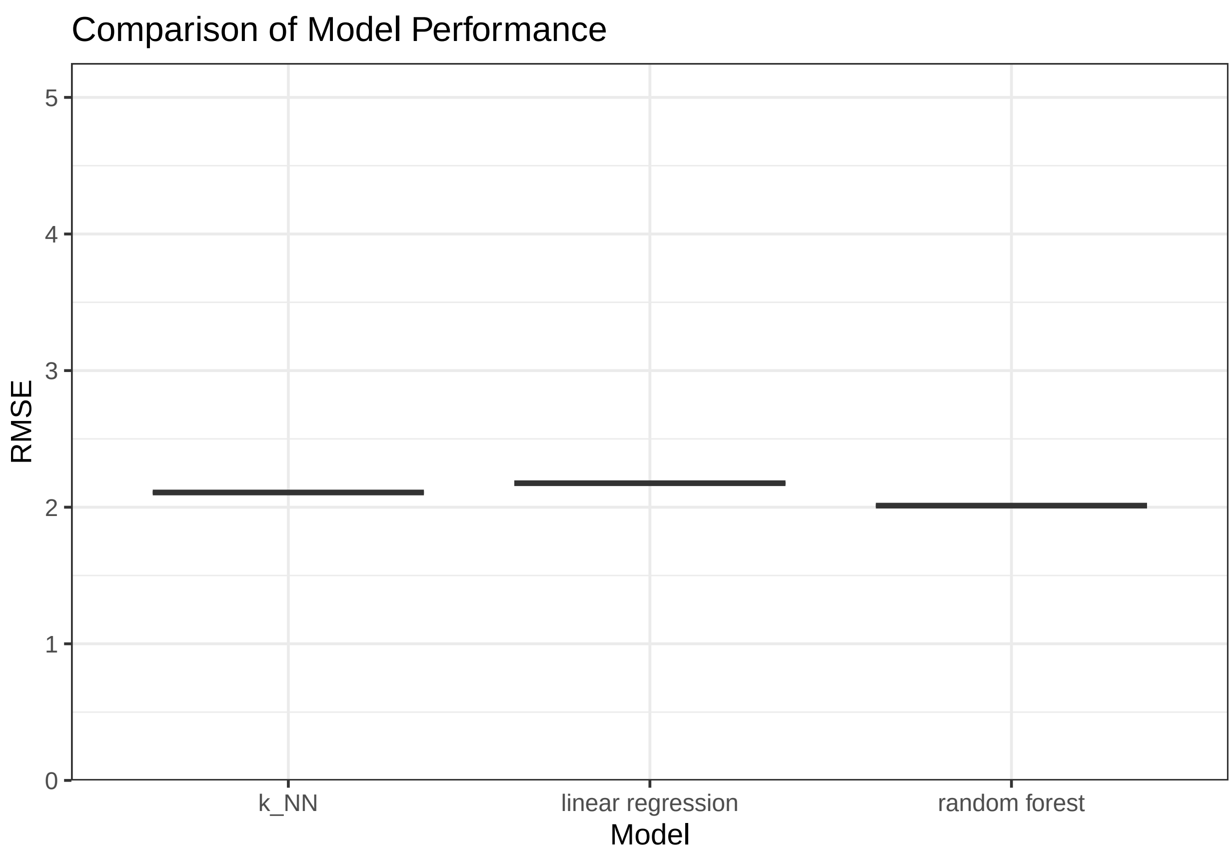
```r
rf_metrics <- rf_res |>
  collect_metrics() |>
  filter(.metric == "rmse") |>
  filter(mtry == 2 & min_n == 3) |>
  mutate(model = "random forest") |>
  arrange(mean)

randFor_rmse = 1.854144
```

Summarizing Results

```r
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      2.01
## 2 k_NN               2.11
## 3 linear regression  2.18
```

```r
# 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()
```



```r
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$differences <- rf_pred$.pred - dat_test$value

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

```
## [1] 1.750998
```

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

```r
dat_test |>
  group_by(state) |>
  summarize(avg = mean(differences), n = n()) |>
  arrange(abs(avg))
```

```
## # A tibble: 48 x 3
##    state              avg     n
##    <chr>            <dbl> <int>
##  1 New York       0.00224     6
##  2 Massachusetts -0.0424      3
##  3 Arkansas      -0.0962      2
##  4 Illinois       0.100       9
##  5 Connecticut    0.103       6
##  6 Missouri      -0.150       3
##  7 South Dakota   0.221       3
##  8 Indiana       -0.249      15
##  9 Oklahoma      -0.256       2
## 10 Washington    -0.260       3
## # i 38 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.