Report 2: Predicting Ambient Air Pollution Concentrations Across the Continental US

Isobel Bodefeld

2023-12-1

Introduction

In this report, I aim to make predictions about ambient air pollution concentrations in the United States. To do so, I will utilize three models: linear regression, k-Nearest Neighbors, and Random Forest. These models can be used on continuous variables which is why I chose them for this report.

Models Chosen

Linear regression is a common approach in statistical modeling and can be useful in describing direction and strength of a linear relationship between two variables. K-Nearest Neighbors approximates the association between independent predictor variables and continuous outcomes by averaging the observations in the same neighborhood. Random Forest can be helpful in understanding complex, non-linear relationships and works through creating multiple decision trees during training. This model is non-parametric, requiring less pre-processing than models like linear regression.

Loading in the Data

```
library(tidyverse)

dat <- read_csv("https://github.com/rdpeng/stat322E_public/raw/main/data/pm25_data.csv.gz")

write_csv(dat, '~/SDS 322E Data Science/Project 2/data.csv')
```

Wrangling

In the following code, I select the predictors that I want to use in my models and save it in a new dataframe ‘selected_dat’. I then pivot my data in to a long format so that I can easily make scatterplots of each predictor later. I also find summary statistics for each predictor to get a sense of the data.

```
# list of selected predictors
selected_predictors <- c('CMAQ', 'aod', 'popdens_county', 'popdens_zcta', 'imp_a10000',
'log_dist_to_prisec',
                         'log_pri_length_10000', 'log_nei_2008_pm25_sum_10000', 'log_nei
_2008_pm10_sum_10000')

# add outcome variable 'value' to the list
selected_predictors <- c('id', selected_predictors, 'value')

# create new dataframe with only selected predictors and outcome variable from 'dat'
selected_dat <- dat[, selected_predictors]

# reshape data into long format for exploratory anaysis
long_data <- pivot_longer(selected_dat,
                          cols = -c(id, value),
                          names_to = 'predictor',
                          values_to = 'predictor_value')

# find summary statistics
selected_dat %>%
  select(-id) %>%
  summarise(across(where(is.numeric), list(
    mean = ~mean(., na.rm = TRUE),
    sd = ~sd(., na.rm = TRUE),
    median = ~median(., na.rm = TRUE),
    IQR = ~IQR(., na.rm = TRUE),
    min = ~min(., na.rm = TRUE),
    max = ~max(., na.rm = TRUE)
  )))
```

```
## # A tibble: 1 × 60
##   CMAQ_mean CMAQ_sd CMAQ_median CMAQ_IQR CMAQ_min CMAQ_max aod_mean aod_sd
##       <dbl>   <dbl>       <dbl>    <dbl>    <dbl>    <dbl>    <dbl>  <dbl>
## 1      8.41    2.97        8.62     3.71     1.63     23.1     43.7   19.6
## # ℹ 52 more variables: aod_median <dbl>, aod_IQR <dbl>, aod_min <dbl>,
## #   aod_max <dbl>, popdens_county_mean <dbl>, popdens_county_sd <dbl>,
## #   popdens_county_median <dbl>, popdens_county_IQR <dbl>,
## #   popdens_county_min <dbl>, popdens_county_max <dbl>,
## #   popdens_zcta_mean <dbl>, popdens_zcta_sd <dbl>, popdens_zcta_median <dbl>,
## #   popdens_zcta_IQR <dbl>, popdens_zcta_min <dbl>, popdens_zcta_max <dbl>,
## #   imp_a10000_mean <dbl>, imp_a10000_sd <dbl>, imp_a10000_median <dbl>, …
```

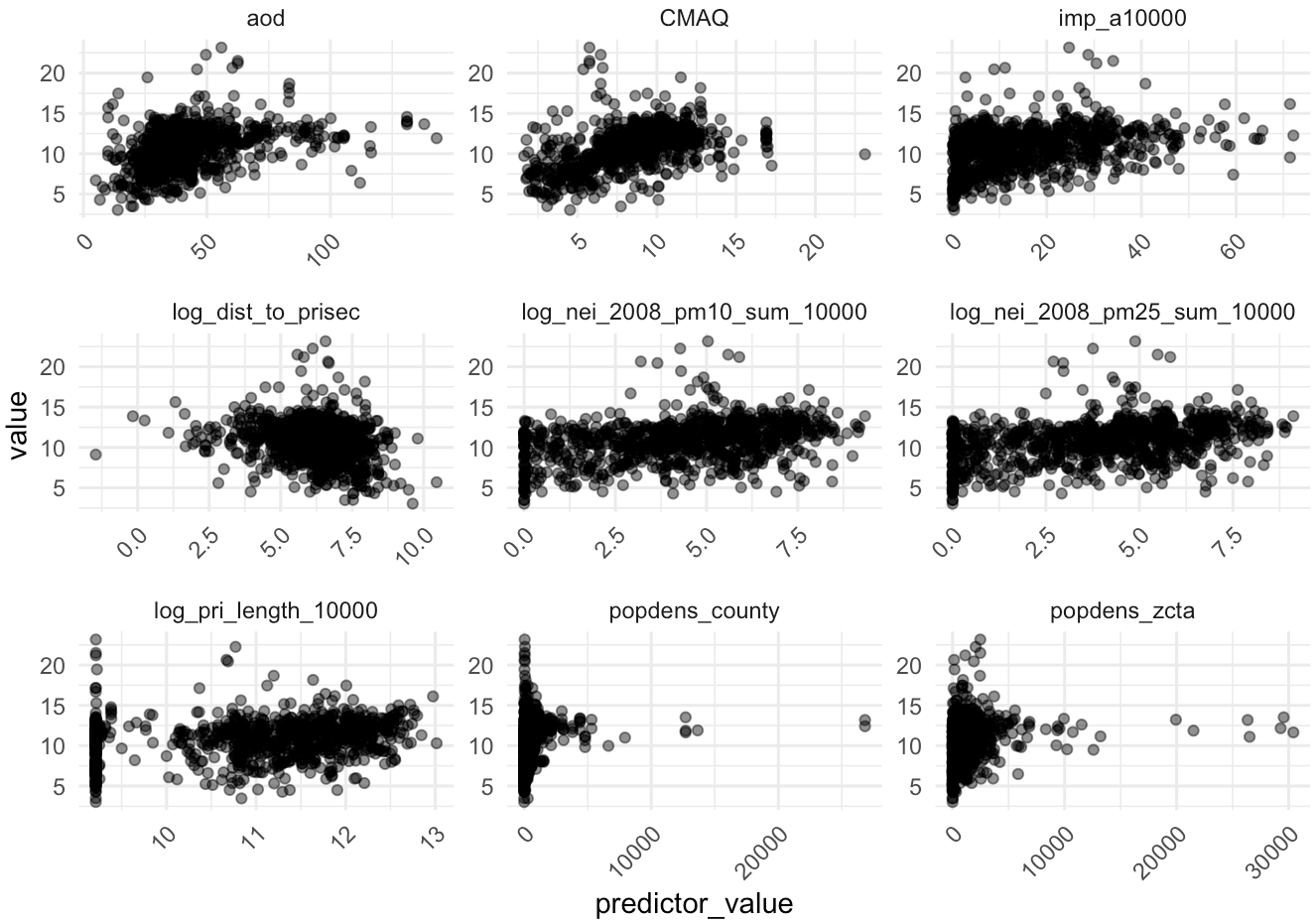
Data & Predictors

I will be using a dataset containing annual average concentrations of fine particulate matter (PM2.5) across the US Environmental Protection Agency's monitoring network in the continental US in addition to many predictor variables. The predictors used in my models include CMAQ, aod, popdens_county, popdens_zcta, imp_a10000, log_dist_to_prisec, log_pri_length10000, log_nei_2008_pm25_sum_10000, and log_nei_2008_pm10_sum_10000.

CMAQ and aod are the predictors of note. CMAQ represents predictions from a numerical computer model and aod represents the “aerosol optical depth”, which is measured from satellites and is related to the amount of pollution near the surface developed by the EPA, which provides meaningful information in predicting air pollution concentrations. I also incorporated other predictors related to development of the area surrounding the monitors.

Exploratory Analysis

```
# make scatterplot between each predictor and outcome (value)
ggplot(long_data, aes(x = predictor_value, y = value)) +
  geom_point(alpha = 0.5) +
  facet_wrap(~ predictor, scales = "free") +  # facet by each predictor
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Noteworthy points about the scatterplots: - Heteroscedasticity for imp_a10000, log_dist_to_prisec, log_pri_length_10000, popdens_county, and popdens_zcta. Linear regression assumes homoscedasticity and the reliability of the standard errors associated with the model estimates would otherwise be impacted.

- log_nei_2008_pm25_sum_10000 and log_nei_2008_pm10_sum_10000 are similar predictor variables and follow a very similar shape and spread, so it’s possible that there is redundancy. They looks to follow a moderate linear relationship that is neither positive or negative.
- Points in aod seem to cluster around the 25-60 range, points in CMAQ seem to be more abundant until 13, and points in log_dist_to_prisec seem to cluster around 4-8. This may suggest subgroups and that kNN would be useful.

Expectations of RMSE Performance

For each of the models, I expect the RMSE value to be lower, say less than 2. After doing exploratory data analysis, I don't believe there are too many factors (outliers, missing data, or gaps) that would cause the model perform very inaccurately. This is also because the outcome variable 'value' and the range are not too large with a minimum of 3.02381 and maximum of 23.16078 (considering the context of the scale).

I do not predict the linear regression model to be the best model due of the lack of homoscedasticity across all predictors in the scatterplots, an assumption of the model. However, I do predict Random Forest to best predict the new data since it can understand more complex, nonlinear data.

Training and Testing Data

Creating two data frames so I have a training dataset and a testing dataset.

```
library(rsample)
set.seed(2351)

# splitting into training and test datasets
split_dat <- selected_dat %>%
  initial_split(prop = 0.75)

train_dat <- training(split_dat)

test_dat <- testing(split_dat)
```

Model 1: Linear Regression

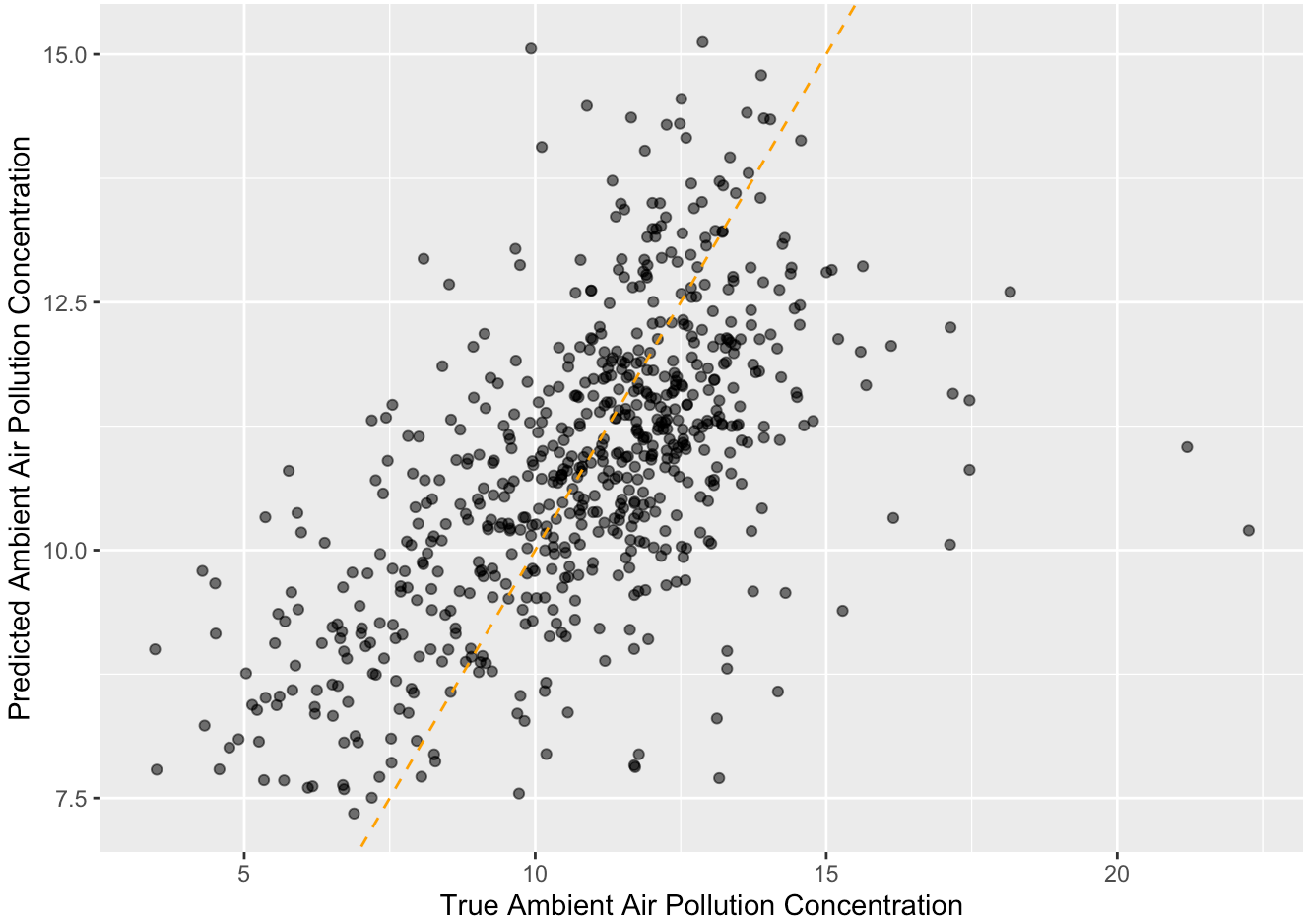
Linear regression is a common approach in statistical modeling and can be useful in describing direction and strength of a linear relationship between two variables. Developed this model by using the training data to create the recipe, model, workflow, and fit using "lm". Predictions were then made using the training data to pre-assess performance and then officially assessed using the testing data.

```r
library(tidymodels)

# recipe
recipe <- train_dat %>%
  recipe(value ~ .) %>%
  step_normalize(all_numeric_predictors(), -all_outcomes())

# linear regression model
lin_reg_model <- linear_reg() %>%
  set_engine("lm") %>%
  set_mode("regression")

# workflow
workflow <- workflow() %>%
  add_recipe(recipe) %>%
  add_model(lin_reg_model)

# fit model
fit <- workflow %>%
  fit(data = train_dat)

wf_fit <- fit %>%
  pull_workflow_fit()

wf_fitted_values <-
  augment(wf_fit$fit, data = train_dat) %>%
  select(value, .fitted:.std.resid)

# make predictions and evaluate model
predictions <- predict(fit, new_data = test_dat) %>%
  bind_cols(test_dat)

# calculate performance metrics like RMSE, R-squared, etc.
metrics <- predictions %>%
  metrics(truth = value, estimate = .pred)

metrics
```

```
## # A tibble: 3 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2.63
## 2 rsq     standard       0.163
## 3 mae     standard       1.85
```

Plotting

```
wf_fitted_values %>%
  ggplot(aes(x = value, y = .fitted)) +
  geom_point(alpha = 0.6) +
  geom_abline(intercept = 0, slope = 1, color = "orange", linetype = "dashed") +
  labs( x = "True Ambient Air Pollution Concentration", y = "Predicted Ambient Air Pollu
tion Concentration")
```



Final Assessment Using Testing Data

```
overallfit <- workflow %>%
  last_fit(split_dat)

collect_metrics(overallfit)
```

```
## # A tibble: 2 × 4
##   .metric .estimator .estimate .config
##   <chr>   <chr>          <dbl> <chr>
## 1 rmse    standard       2.63  Preprocessor1_Model1
## 2 rsq     standard       0.163 Preprocessor1_Model1
```

Model 2: k-Nearest Neighbors

K-Nearest Neighbors approximates the association between independent predictor variables and continuous outcomes by averaging the observations in the same neighborhood. Created the model by making a recipe, mode, and workflow before tuning using the number of neighbors and establishing folds and grids. These model was then fitted and the test data was prepped and extracted.

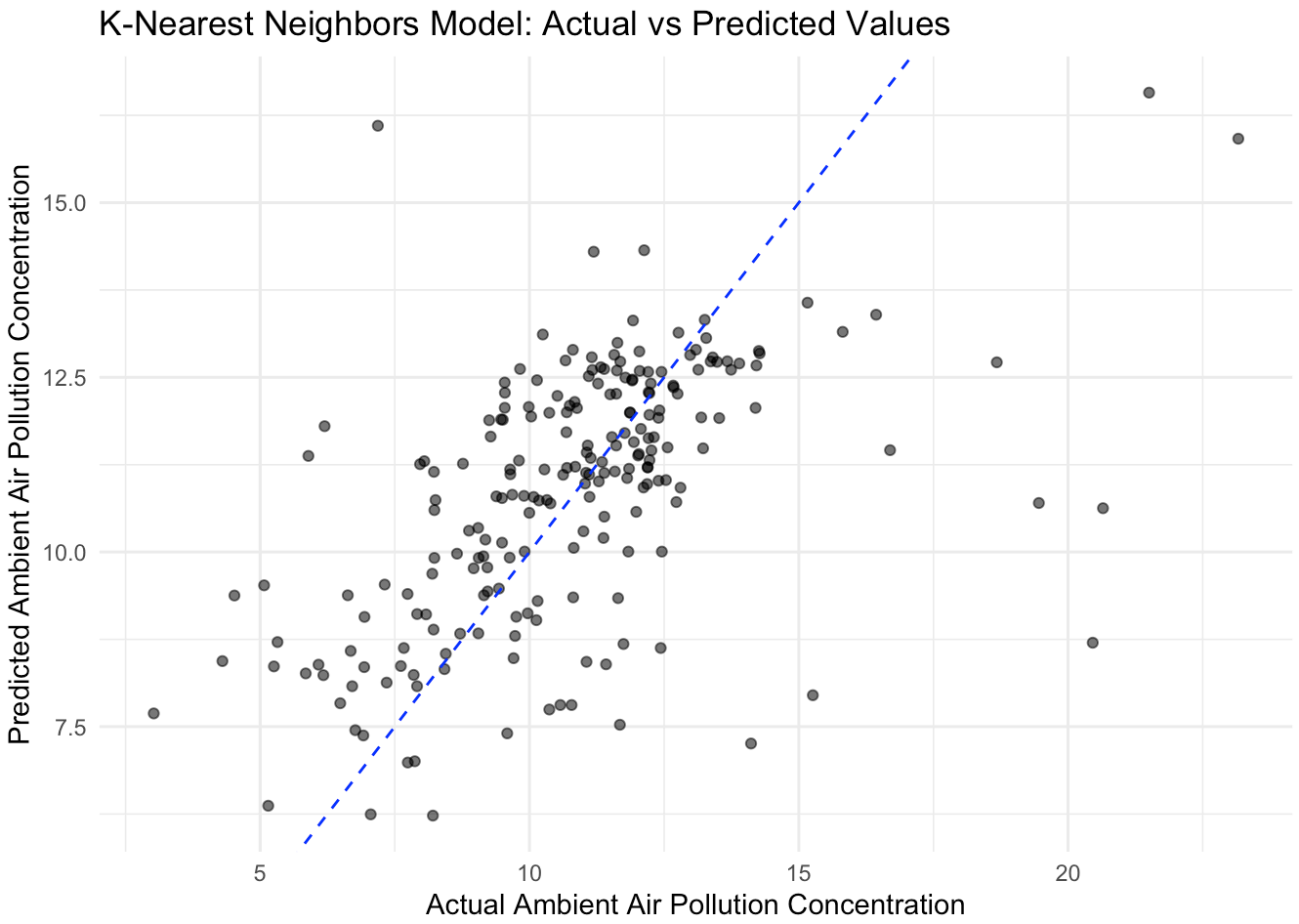
```
# recipe
knn_rec <- train_dat %>%
    recipe(value ~ .)

# model
knn_model <- nearest_neighbor(neighbors = 10) %>%
    set_engine("kknn") %>%
    set_mode("regression")

# workflow
knn_wf <- workflow() %>%
    add_model(knn_model) %>%
    add_recipe(knn_rec)

# tune
model <- nearest_neighbor(neighbors = tune("k")) %>%
    set_engine("kknn") %>%
    set_mode("regression")

wf <- workflow() %>%
    add_model(model) %>%
    add_recipe(knn_rec)

#establish folds and grid
knn_folds <- vfold_cv(train_dat, v = 10)

res <- tune_grid(wf, resamples = knn_folds,
                 grid = tibble(k = c(3, 5, 10, 15, 20, 25)))

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

```
## # A tibble: 5 × 7
##       k .metric .estimator  mean     n std_err .config
##   <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1    15 rmse    standard    1.90    10  0.0963 Preprocessor1_Model4
## 2    20 rmse    standard    1.90    10  0.0959 Preprocessor1_Model5
## 3    10 rmse    standard    1.90    10  0.0968 Preprocessor1_Model3
## 4    25 rmse    standard    1.91    10  0.0946 Preprocessor1_Model6
## 5     5 rmse    standard    1.96    10  0.101  Preprocessor1_Model2
```

```
# fit
knn_fit <- fit(knn_wf, data = train_dat)

# prepped data for testing
prepped_test_dat <- knn_rec %>%
  prep(train_dat) %>%
  bake(new_data = test_dat)

knn_fit %>%
  extract_fit_parsnip() %>%
  augment(new_data = prepped_test_dat) %>%
  summarise(
    rmse = sqrt(mean(.resid ** 2)))
```

```
## # A tibble: 1 × 1
##    rmse
##   <dbl>
## 1  2.39
```

```
# generate predictions using fitted KNN model
predictions <- knn_fit %>%
  predict(new_data = prepped_test_dat) %>%
  bind_cols(prepped_test_dat)

# rename the prediction column
predictions <- predictions %>%
  rename(Predicted = .pred)

# create scatterplot
ggplot(predictions, aes(x = value, y = Predicted)) +
  geom_point(alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, color = "blue", linetype = "dashed") +
  labs(title = "K-Nearest Neighbors Model: Actual vs Predicted Values",
       x = "Actual Ambient Air Pollution Concentration",
       y = "Predicted Ambient Air Pollution Concentration") +
  theme_minimal()
```



Model 3: Random Forest

Random Forest can be helpful in understanding complex, non-linear relationships and works through creating multiple decision trees during training. This model is non-parametric. To do this, I made a recipe, mode, and workflow before establishing folds and grids. Predictions were then made using fitted model and prepped testing data.

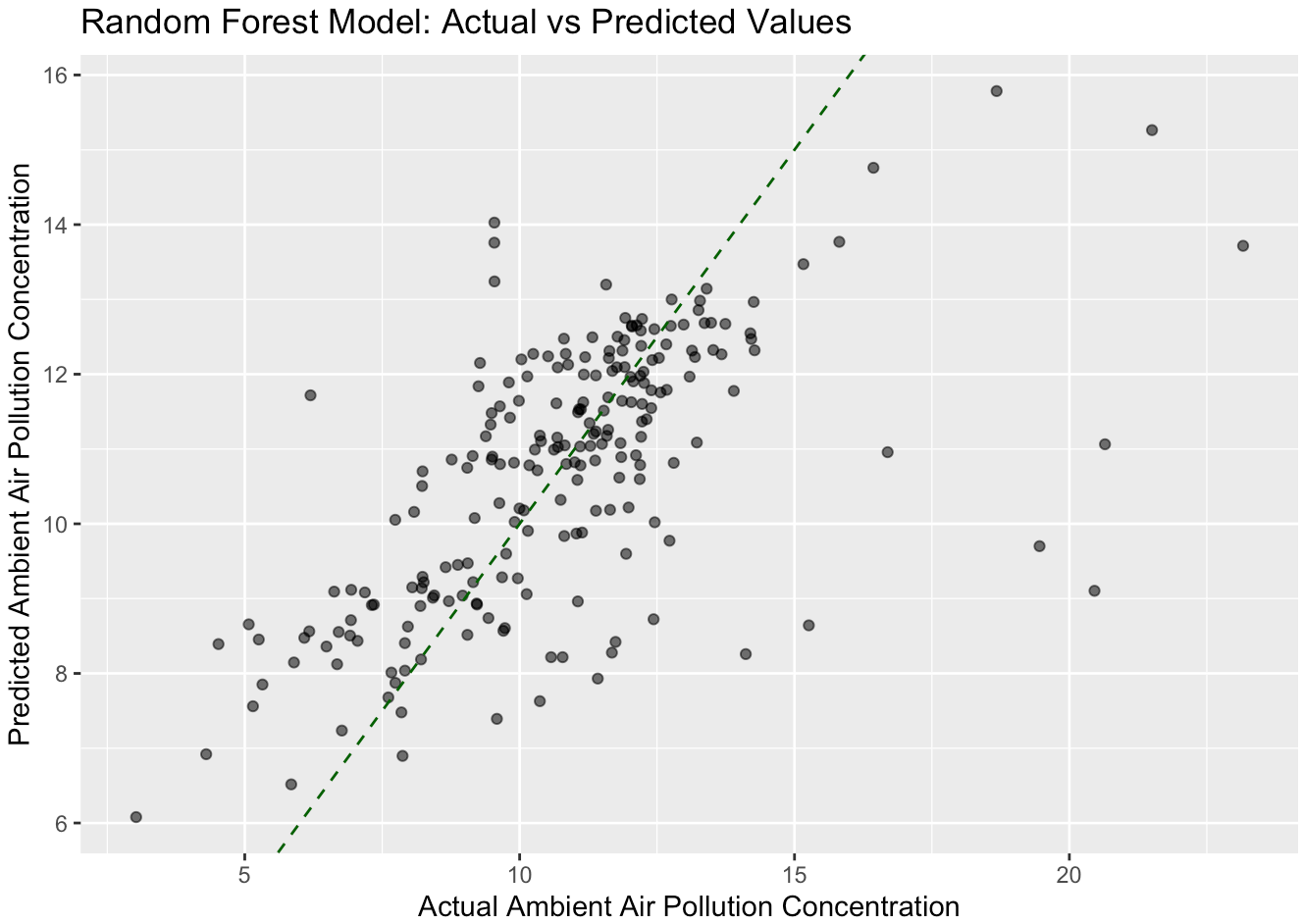
```
# recipe
rf_rec <- train_dat %>%
  recipe(value ~ .)

# model
rf_model <- rand_forest() %>%
  set_engine("randomForest") %>%
  set_mode("regression")

# workflow
rf_wf <- workflow() %>%
  add_model(rf_model) %>%
  add_recipe(rf_rec)

# tune for optimal parameters
model <- rand_forest() %>%
  set_engine("randomForest") %>%
  set_mode("regression")

wf <- workflow() %>%
  add_model(model) %>%
  add_recipe(rf_rec)

# establish folds
rf_folds <- vfold_cv(train_dat, v = 10)

res <- tune_grid(wf, resamples = rf_folds,
                 grid = tibble(trees = c(50, 100, 150)))  # Adjust the range of trees

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

```
## # A tibble: 1 × 6
##   .metric .estimator  mean     n std_err .config
##   <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1 rmse    standard    1.74    10  0.0706 Preprocessor1_Model1
```

```
# fit
rf_fit <- fit(rf_wf, data = train_dat)

prepped_test_dat <- rf_rec %>%
  prep(train_dat) %>%
  bake(new_data = test_dat)

rf_fit %>%
  predict(new_data = prepped_test_dat) %>%
  bind_cols(prepped_test_dat) %>%
  metrics(truth = value, estimate = .pred) %>%
  filter(.metric == "rmse")
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        2.20
```

```
# plot actual vs predicted values as a scatterplot
predictions <- predict(rf_fit, new_data = prepped_test_dat)
plot_dat <- data.frame(
  actual = test_dat$value,
  predicted = predictions$.pred)

ggplot(data = plot_dat, aes(x = actual, y = predicted)) +
  geom_point(alpha = 0.6) +
  geom_abline(intercept = 0, slope = 1, color = "darkgreen", linetype = "dashed") +
  labs(x = "Actual Ambient Air Pollution Concentration", y = "Predicted Ambient Air Poll
ution Concentration") +
  ggtitle("Random Forest Model: Actual vs Predicted Values")
```



Comparing Results

```
lin_reg_rmse <- 2.63
knn_rmse <- 2.39
ran_for_rmse <- 2.20

# creating a summary table
model_comparison <- tibble(
  Model = c("Linear Regression", "K-Nearest Neighbors", "Random Forest"),
  RMSE = c(lin_reg_rmse, knn_rmse, ran_for_rmse)
)

model_comparison
```

```
## # A tibble: 3 × 2
##   Model                RMSE
##   <chr>               <dbl>
## 1 Linear Regression    2.63
## 2 K-Nearest Neighbors  2.39
## 3 Random Forest        2.2
```

RMSE Values Across Models: Linear Regression: 2.63 k-Nearest Neighbors: 2.39 Random Forest: 2.20

Looking at the results of RMSE values across the three models, Random Forest seemed to perform the best on the data because it has the lowest value.

Discussion

Primary Questions:

1. Based on test set performance, at what locations does your model give predictions that are closest and furthest from the observed values? What do you hypothesize are the reasons for the good or bad performance at these locations?

```
predictions <- rf_fit %>% predict(new_data = prepped_test_dat)
errors <- abs(predictions - test_dat$value)  # Absolute prediction errors

# combine errors with test data
error_df <- cbind(test_dat, PredictionError = errors)

# identify locations with closest and furthest predictions
closest_predictions <- error_df %>% arrange(errors) %>% head()
furthest_predictions <- error_df %>% arrange(desc(errors)) %>% head()

# corresponding rows in test_dat
closest_locations <- closest_predictions$id
furthest_locations <- furthest_predictions$id

# extract corresponding rows from test_dat
test_dat_closest <- dat[dat$id %in% closest_locations, ]
test_dat_closest
```

```
## # A tibble: 6 × 50
##       id value  fips   lat    lon state      county  city   CMAQ  zcta zcta_area
##    <dbl> <dbl> <dbl> <dbl>  <dbl> <chr>      <chr>   <chr> <dbl> <dbl>     <dbl>
## 1  6045.  8.21  6045  39.2 -123.  California Mendoc… Ukiah  2.91 95482 805163258
## 2 17089. 10.8  17089  42.1  -88.3 Illinois   Kane    Elgin 10.4  60120  42961291
## 3 19197.  9.15 19197  42.7  -93.7 Iowa       Wright  Clar…  6.45 50525 361364714
## 4 21059. 12.0  21059  37.8  -87.1 Kentucky   Daviess Not … 11.8  42303 124084216
## 5 27123. 11.1  27123  45.0  -93.1 Minnesota  Ramsey  St. …  7.99 55101   2109625
## 6 51107. 11.5  51107  39.0  -77.5 Virginia   Loudoun Not …  9.19 20147  51473450
## # ℹ 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>, …
```

```
test_dat_furthest <- dat[dat$id %in% furthest_locations, ]
test_dat_furthest
```

```
## # A tibble: 6 × 50
##      id value  fips   lat   lon state      county city      CMAQ  zcta zcta_area
##   <dbl> <dbl> <dbl> <dbl> <dbl> <chr>      <chr>  <chr>    <dbl> <dbl>     <dbl>
## 1 4021.  19.5  4021  33.0 -112. Arizona    Pinal  Maricopa 11.5  85138 191824823
## 2 6007.  20.5  6007  39.8 -122. California Butte  Not in …  5.36 95926  19815814
## 3 6029.  21.5  6029  35.4 -119. California Kern   Bakersf…  5.75 93304  19670450
## 4 6029.  23.2  6029  35.3 -119. California Kern   Bakersf…  5.75 93304  19670450
## 5 6089.  15.3  6089  40.5 -122. California Shasta Redding   3.43 96001 227797024
## 6 6107.  20.6  6107  36.3 -119. California Tulare Visalia   6.58 93292 281391266
## # ℹ 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>, …
```

The locations that my model gives predictions that are closest from the observed values are from the states California, Illinois, Iowa, Kentucky, Minnesota, and Virginia. The locations where the predictions are furthest from the observed values are from the states Arizona and many from California. I hypothesize that a few predictors may play a role in the performance including populations density and zcta. However, the top results for each aren't all similar so it is hard to make an opinion on this.

2. What variables might predict where your model performs well or not? For example, are their regions of the country where the model does better or worse? Are there variables that are not included in this dataset that you think might improve the model performance if they were included in your model?

Variables related to the population density like popdens_county and popdens_zcta could indicate where the performs well or not. This is because there was higher accuracy when the populations was higher/denser compared to their less populated/sparse counterparts. This is due to the amount of data present in the overall/training dataset that biases which types perform best.

3. There is interest in developing more cost-effect approaches to monitoring air pollution on the ground. Two candidates for replacing the use of ground-based monitors are numerical models like CMAQ and satellite-based observations such as AOD. How well do CMAQ and AOD predict ground-level concentrations of PM2.5? How does the prediction performance of your model change when CMAQ or aod are included (or not included) in the model?

```
# list of selected predictors
selected_predictors_q3 <- c('popdens_county', 'popdens_zcta', 'imp_a10000', 'log_dist_to
_prisec',
                            'log_pri_length_10000', 'log_nei_2008_pm25_sum_10000', 'log_nei
_2008_pm10_sum_10000')

# add outcome variable 'value' to the list
selected_predictors_q3 <- c('id', selected_predictors_q3, 'value')

# create new dataframe with only selected predictors and outcome variable from 'dat'
selected_dat_q3 <- dat[, selected_predictors_q3]

# splitting into training and test datasets
split_dat_q3 <- selected_dat_q3 %>%
  initial_split(prop = 0.75)

train_dat_q3 <- training(split_dat_q3)

test_dat_q3 <- testing(split_dat_q3)

# Creating recipe
rf_rec <- train_dat_q3 %>%
  recipe(value ~ .)

# Creating Random Forest model
rf_model <- rand_forest() %>%
  set_engine("randomForest") %>%
  set_mode("regression")

# Creating workflow
rf_wf <- workflow() %>%
  add_model(rf_model) %>%
  add_recipe(rf_rec)

# Tuning for optimal parameters
model <- rand_forest() %>%
  set_engine("randomForest") %>%
  set_mode("regression")

wf <- workflow() %>%
  add_model(model) %>%
  add_recipe(rf_rec)

rf_folds <- vfold_cv(train_dat, v = 10)

res <- tune_grid(wf, resamples = rf_folds,
                 grid = tibble(trees = c(50, 100, 150)))

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

```
## # A tibble: 1 × 6
##   .metric .estimator  mean     n std_err .config
##   <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1 rmse    standard    1.94    10  0.0717 Preprocessor1_Model1
```

```
rf_fit <- fit(rf_wf, data = train_dat)

prepped_test_dat <- rf_rec %>%
  prep(train_dat) %>%
  bake(new_data = test_dat)

rf_fit %>%
  predict(new_data = prepped_test_dat) %>%
  bind_cols(prepped_test_dat) %>%
  metrics(truth = value, estimate = .pred) %>%
  filter(.metric == "rmse")
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        2.44
```

4. The dataset here did not include data from Alaska or Hawaii. Do you think your model will perform well or not in those two states? Explain your reasoning.

I don't think the model will perform well in these two states due to big differences in climate and geographical region. The 48 states that the model has data on is of the continental US shares a more similar geographical area. Alaska and Hawaii are ve4ry removed from the mainland and the model may perform differently due to the differences in predictors here.

Reflection

Reflect on the process of conducting this project. What was challenging, what have you learned from the process itself?

After painfully learning throughout this semester, I learned that a good and responsible process of conducting a data project is crucial. I began with looking at the data itself to get a better understanding of what I'm working on and how I should handle it. This then made me make better decisions for how to wrangle and decide which predictors to use for the models I chose.

With my models, linear regression, kNN, and random forest, I found a lot of my challenges to be related to how to pre-process and tune it so that the model works accurately. This was very tedious and Googling (esp StackOverflow) became my best friend.

Through these challenges, I learned a lot about implementing these models in code, especially Random Forest. Random forest peaked my curiosity most becaue I have seen it used in other data science projects I've assisted with as well as technical assessments during recruiting season. Before, I didn't really understand how they worked or how to create it, but this project made it up close and personal.

Reflect on the performance of your final prediction model. Did it perform as well as you originally expected? If not, why do you think it didn't perform as well?

The final model did perform as well as I thought. I expected an RMSE around 1-5, lower of this range if anything. I think this was logical based on the amount of data in the dataset with values that didn't range greatly. Even between the 3 models, the RMSE values were not extremely far off from each other. To perform better, I am sure more could have been done during the data pre-processing step to reduce extraneous info or values that cause the predictive model to become biased. Machine learning is a tricky science to get the hang of!

Acknowledgememts

I can't take credit for this assignment without acknowledging our TA Soumyabrata Bose and Professor Steven Rashin. Bose helped me a lot in understanding the steps necessary to solve the problems I faced and provided resources so I could figure it out myself. In addition, Rashin provided thorough context for the assignment and a high-level plan to guide us through it. Thank you.