DPA

Syed Wali Uddin Quadri

2024-04-19

# Load required libraries  
library(readr)

## Warning: package 'readr' was built under R version 4.3.3

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.3.3

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

# Load the data  
data <- read\_csv("C:/Users/Nusrath shaheen/OneDrive/Desktop/DPA/Project/Final\_COPD\_encoded.csv")

## New names:  
## Rows: 3080 Columns: 23  
## ── Column specification  
## ──────────────────────────────────────────────────────── Delimiter: "," chr  
## (5): State...1, County, Encoded\_Poverty, State...22, Area\_Name dbl (16): Carbon  
## monoxide (CO), Nitrogen dioxide (NO2), Ozone, PM10 Total 0-... num (2): Average  
## Income, Unemployed  
## 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.  
## • `State` -> `State...1`  
## • `State` -> `State...22`

head(data)

## # A tibble: 6 × 23  
## State...1 County `Carbon monoxide (CO)` `Nitrogen dioxide (NO2)` Ozone  
## <chr> <chr> <dbl> <dbl> <dbl>  
## 1 Alabama Autauga 0.165 14.6 0.0457  
## 2 Alabama Baldwin 0.165 14.6 0.0439  
## 3 Alabama Barbour 0.165 14.6 0.0457  
## 4 Alabama Bibb 0.165 14.6 0.0457  
## 5 Alabama Blount 0.165 14.6 0.0457  
## 6 Alabama Bullock 0.165 14.6 0.0457  
## # ℹ 18 more variables: `PM10 Total 0-10um STP` <dbl>,  
## # `PM10-2.5 - Local Conditions` <dbl>, `PM2.5 - Local Conditions` <dbl>,  
## # `Sulfur PM2.5 LC` <dbl>, `Sulfur dioxide (SO2)` <dbl>,  
## # `Average Income` <dbl>, Percent\_COPD <dbl>, `Male 18-44` <dbl>,  
## # `Male 45-64` <dbl>, `Male 65+` <dbl>, `Female 18-44` <dbl>,  
## # `Female 45-64` <dbl>, `Female 65+` <dbl>, Poverty <dbl>,  
## # Encoded\_Poverty <chr>, Unemployed <dbl>, State...22 <chr>, …

# Feature Engineering  
# Age Group Aggregation  
data$Male\_Total <- rowSums(data[, c("Male 18-44", "Male 45-64", "Male 65+")])  
data$Female\_Total <- rowSums(data[, c("Female 18-44", "Female 45-64", "Female 65+")])

# Air Quality Index (AQI)  
data$AQI <- rowSums(data[, c("Carbon monoxide (CO)", "Nitrogen dioxide (NO2)", "Ozone", "PM10 Total 0-10um STP", "PM10-2.5 - Local Conditions", "PM2.5 - Local Conditions", "Sulfur PM2.5 LC", "Sulfur dioxide (SO2)"  
)])

# Convert Average\_Income column to numeric  
# Check if Average Income column exists  
if("Average Income" %in% colnames(data)) {  
 # Convert Average Income column to numeric  
 data$Average\_Income <- as.numeric(as.character(data$"Average Income"))  
   
 # Create Income\_Level based on Average\_Income  
 data$Income\_Level <- cut(data$Average\_Income, breaks = c(0, 30000, 60000, Inf), labels = c("Low", "Medium", "High"), right = FALSE)  
} else {  
 print("Column 'Average Income' not found in the dataset.")  
}

# Now, create Income\_Level based on Average\_Income  
data$Income\_Level <- cut(data$Average\_Income, breaks = c(0, 30000, 60000, Inf), labels = c("Low", "Medium", "High"), right = FALSE)

# Calculate mean, median, maximum, and minimum for the Unemployed column  
mean\_unemployed <- mean(data$Unemployed, na.rm = TRUE)  
median\_unemployed <- median(data$Unemployed, na.rm = TRUE)  
max\_unemployed <- max(data$Unemployed, na.rm = TRUE)  
min\_unemployed <- min(data$Unemployed, na.rm = TRUE)  
  
# Print the results  
print(paste("Mean Unemployed:", mean\_unemployed))

## [1] "Mean Unemployed: 4154.41501462463"

print(paste("Median Unemployed:", median\_unemployed))

## [1] "Median Unemployed: 771"

print(paste("Maximum Unemployed:", max\_unemployed))

## [1] "Maximum Unemployed: 609830"

print(paste("Minimum Unemployed:", min\_unemployed))

## [1] "Minimum Unemployed: 4"

# Calculate quartiles  
quartiles <- quantile(data$Unemployed, probs = c(0, 0.25, 0.5, 0.75, 0.9, 1), na.rm = TRUE)  
  
# Categorize unemployed column based on quartiles  
data$Unemployed\_Level <- cut(data$Unemployed,  
 breaks = c(min(data$Unemployed), quartiles),  
 labels = c("Very Low", "Low", "Moderate", "High", "Very High"),  
 include.lowest = TRUE)

head(data)

## # A tibble: 6 × 29  
## State...1 County `Carbon monoxide (CO)` `Nitrogen dioxide (NO2)` Ozone  
## <chr> <chr> <dbl> <dbl> <dbl>  
## 1 Alabama Autauga 0.165 14.6 0.0457  
## 2 Alabama Baldwin 0.165 14.6 0.0439  
## 3 Alabama Barbour 0.165 14.6 0.0457  
## 4 Alabama Bibb 0.165 14.6 0.0457  
## 5 Alabama Blount 0.165 14.6 0.0457  
## 6 Alabama Bullock 0.165 14.6 0.0457  
## # ℹ 24 more variables: `PM10 Total 0-10um STP` <dbl>,  
## # `PM10-2.5 - Local Conditions` <dbl>, `PM2.5 - Local Conditions` <dbl>,  
## # `Sulfur PM2.5 LC` <dbl>, `Sulfur dioxide (SO2)` <dbl>,  
## # `Average Income` <dbl>, Percent\_COPD <dbl>, `Male 18-44` <dbl>,  
## # `Male 45-64` <dbl>, `Male 65+` <dbl>, `Female 18-44` <dbl>,  
## # `Female 45-64` <dbl>, `Female 65+` <dbl>, Poverty <dbl>,  
## # Encoded\_Poverty <chr>, Unemployed <dbl>, State...22 <chr>, …

# Remove unnecessary columns  
# Exclude columns with special characters from the subset  
data <- subset(data, select = -c(`Carbon monoxide (CO)`, `Nitrogen dioxide (NO2)`, `Ozone`, `PM10 Total 0-10um STP`, `PM10-2.5 - Local Conditions`, `PM2.5 - Local Conditions`, `Sulfur PM2.5 LC`, `Sulfur dioxide (SO2)`))

# Remove rows with missing values  
data <- na.omit(data)

# Load required library  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:randomForest':  
##   
## combine

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# Create dummy variables for Income\_Level  
income\_dummies <- model.matrix(~ Income\_Level - 1, data = data)  
  
# Create dummy variables for Unemployed\_Level  
unemployed\_dummies <- model.matrix(~ Unemployed\_Level - 1, data = data)  
  
# Create dummy variables for Encoded\_Poverty  
poverty\_dummies <- model.matrix(~ Encoded\_Poverty - 1, data = data)  
  
# Combine original data with dummy variables  
data\_encoded <- cbind(data, income\_dummies, unemployed\_dummies, poverty\_dummies)  
  
# Remove original categorical variables  
data\_encoded <- select(data\_encoded, -c(Income\_Level, Unemployed\_Level, Encoded\_Poverty))  
  
# Print the first few rows of the encoded data  
head(data\_encoded)

## State...1 County Average Income Percent\_COPD Male 18-44 Male 45-64 Male 65+  
## 1 Alabama Autauga 45151 6.8 6 10.3 15.1  
## 2 Alabama Baldwin 51230 6.5 6 10.3 15.1  
## 3 Alabama Barbour 37111 9.7 6 10.3 15.1  
## 4 Alabama Bibb 34938 8.6 6 10.3 15.1  
## 5 Alabama Blount 38133 7.8 6 10.3 15.1  
## 6 Alabama Bullock 30495 9.8 6 10.3 15.1  
## Female 18-44 Female 45-64 Female 65+ Poverty Unemployed State...22  
## 1 7.1 11.6 16.4 6296 1402 AL  
## 2 7.1 11.6 16.4 25526 6086 AL  
## 3 7.1 11.6 16.4 5089 667 AL  
## 4 7.1 11.6 16.4 4204 632 AL  
## 5 7.1 11.6 16.4 6992 1126 AL  
## 6 7.1 11.6 16.4 2764 291 AL  
## Area\_Name Male\_Total Female\_Total AQI Average\_Income  
## 1 Autauga County, AL 31.4 35.1 52.97888 45151  
## 2 Baldwin County, AL 31.4 35.1 51.89315 51230  
## 3 Barbour County, AL 31.4 35.1 52.97888 37111  
## 4 Bibb County, AL 31.4 35.1 52.97888 34938  
## 5 Blount County, AL 31.4 35.1 52.97888 38133  
## 6 Bullock County, AL 31.4 35.1 52.97888 30495  
## Income\_LevelLow Income\_LevelMedium Income\_LevelHigh Unemployed\_LevelVery Low  
## 1 0 1 0 0  
## 2 0 1 0 0  
## 3 0 1 0 0  
## 4 0 1 0 0  
## 5 0 1 0 0  
## 6 0 1 0 1  
## Unemployed\_LevelLow Unemployed\_LevelModerate Unemployed\_LevelHigh  
## 1 0 1 0  
## 2 0 0 1  
## 3 1 0 0  
## 4 1 0 0  
## 5 0 1 0  
## 6 0 0 0  
## Unemployed\_LevelVery High Encoded\_PovertyHigh Encoded\_PovertyLow  
## 1 0 0 1  
## 2 0 0 0  
## 3 0 0 1  
## 4 0 0 1  
## 5 0 0 1  
## 6 0 0 1  
## Encoded\_PovertyMedium Encoded\_PovertyVery High Encoded\_PovertyVery Low  
## 1 0 0 0  
## 2 1 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## 6 0 0 0

# Output the modified dataset to a CSV file  
write.csv(data\_encoded, file = "final ml data2.csv", row.names = FALSE)

features <- c("Income\_LevelLow", "Income\_LevelMedium", "Income\_LevelHigh", "Unemployed\_LevelVery Low",   
 "Unemployed\_LevelLow", "Unemployed\_LevelModerate", "Unemployed\_LevelHigh", "Unemployed\_LevelVery High",   
 "AQI", "Encoded\_PovertyVery Low", "Encoded\_PovertyLow", "Encoded\_PovertyMedium",   
 "Encoded\_PovertyHigh", "Encoded\_PovertyVery High")  
  
features2 <- c("AQI")  
  
features3 <- c("Income\_LevelLow", "Income\_LevelMedium", "Income\_LevelHigh", "AQI")  
  
features4 <- c( "Unemployed\_LevelVery Low",   
 "Unemployed\_LevelLow", "Unemployed\_LevelModerate", "Unemployed\_LevelHigh", "Unemployed\_LevelVery High", "AQI")  
  
features5 <- c("Income\_LevelLow", "Income\_LevelMedium", "Income\_LevelHigh", "Encoded\_PovertyVery Low", "Encoded\_PovertyLow", "Encoded\_PovertyMedium",   
 "Encoded\_PovertyHigh", "Encoded\_PovertyVery High","AQI")  
  
features6 <- c("Income\_LevelLow", "Income\_LevelMedium", "Income\_LevelHigh", "Unemployed\_LevelVery Low",   
 "Unemployed\_LevelLow", "Unemployed\_LevelModerate", "Unemployed\_LevelHigh", "Unemployed\_LevelVery High")

# Split data into training and testing sets  
set.seed(36)  
trainIndex <- createDataPartition(data\_encoded$Percent\_COPD, p = 0.8, list = FALSE)  
train\_data <- data\_encoded[trainIndex, ]  
test\_data <- data\_encoded[-trainIndex, ]

# Train Linear Regression model  
lm\_model <- train(Percent\_COPD ~ ., data = train\_data[, c(features, "Percent\_COPD")], method = "lm")

library(gbm)

## Warning: package 'gbm' was built under R version 4.3.3

## Loaded gbm 2.1.9

## This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.com/gbm-developers/gbm3

# Train Gradient Boosting Machine (GBM) model  
gbm\_model <- gbm(Percent\_COPD ~ ., data = train\_data[, c(features, "Percent\_COPD")], distribution = "gaussian", n.trees = 100, interaction.depth = 4, shrinkage = 0.01)

# Make predictions  
lm\_predictions <- predict(lm\_model, newdata = test\_data)  
gbm\_predictions <- predict(gbm\_model, newdata = test\_data[, features], n.trees = 100)

# Evaluate models  
lm\_mae <- mean(abs(test\_data$Percent\_COPD - lm\_predictions))  
  
lm\_mse <- mean((test\_data$Percent\_COPD - lm\_predictions)^2)  
  
lm\_rmse <- sqrt(mean((test\_data$Percent\_COPD - lm\_predictions)^2))  
  
lm\_r\_squared <- cor(test\_data$Percent\_COPD, lm\_predictions)^2

# Calculate Mean Absolute Error (MAE) for GBM model  
gbm\_mae <- mean(abs(test\_data$Percent\_COPD - gbm\_predictions))  
  
# Calculate Mean Squared Error (MSE) for GBM model  
gbm\_mse <- mean((test\_data$Percent\_COPD - gbm\_predictions)^2)  
  
# Calculate Root Mean Squared Error (RMSE) for GBM model  
gbm\_rmse <- sqrt(mean((test\_data$Percent\_COPD - gbm\_predictions)^2))  
  
# Calculate R-squared (R²) for GBM model  
gbm\_r\_squared <- cor(test\_data$Percent\_COPD, gbm\_predictions)^2

# Print evaluation metrics  
cat("Linear Regression Metrics:\n")

## Linear Regression Metrics:

cat("MAE:", lm\_mae, "\n")

## MAE: 1.2278

cat("MSE:", lm\_mse, "\n")

## MSE: 2.564139

cat("RMSE:", lm\_rmse, "\n")

## RMSE: 1.601293

cat("R-squared:", lm\_r\_squared, "\n")

## R-squared: 0.0634528

cat("\nGBMs Forest Metrics:\n")

##   
## GBMs Forest Metrics:

cat("MAE:", gbm\_mae, "\n")

## MAE: 1.090981

cat("MSE:", gbm\_mse, "\n")

## MSE: 1.959407

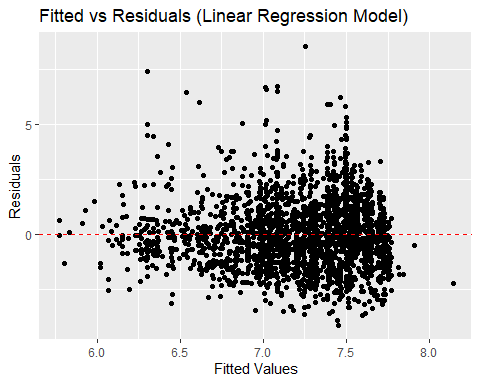
cat("RMSE:", gbm\_rmse, "\n")

## RMSE: 1.399788

cat("R-squared:", gbm\_r\_squared, "\n")

## R-squared: 0.3943545

# Load required libraries  
library(ggplot2)  
  
# Create a dataframe with fitted values and residuals  
lm\_resid <- resid(lm\_model$finalModel)  
lm\_fitted <- predict(lm\_model$finalModel)  
lm\_plot\_data <- data.frame(Fitted = lm\_fitted, Residual = lm\_resid)  
  
# Create scatter plot  
lm\_plot <- ggplot(lm\_plot\_data, aes(x = Fitted, y = Residual)) +  
 geom\_point() +  
 geom\_hline(yintercept = 0, linetype = "dashed", color = "red") +  
 labs(x = "Fitted Values", y = "Residuals", title = "Fitted vs Residuals (Linear Regression Model)")  
  
# Print the plot  
print(lm\_plot)



# Calculate residuals for GBM model  
gbm\_resid <- test\_data$Percent\_COPD - gbm\_predictions  
  
# Create a dataframe with fitted values and residuals  
gbm\_plot\_data <- data.frame(Fitted = gbm\_predictions, Residual = gbm\_resid)  
  
# Create scatter plot of fitted values vs residuals  
gbm\_plot <- ggplot(gbm\_plot\_data, aes(x = Fitted, y = Residual)) +  
 geom\_point() +  
 geom\_hline(yintercept = 0, linetype = "dashed", color = "red") +  
 labs(x = "Fitted Values", y = "Residuals", title = "GBM Fitted vs Residuals")  
  
# Print the plot  
print(gbm\_plot)

