Report 3

Advay Vyas

September 18, 2025

Contents

```
library(dplyr)
library(tidyverse)
library(ggplot2)
library(lubridate)
library(patchwork)
library(corrplot)
library(mosaic)
library(moderndive)
library(effectsize)
library(tidyr)
library(caret)
library(purrr)
library(fastDummies)
library(GGally)
```

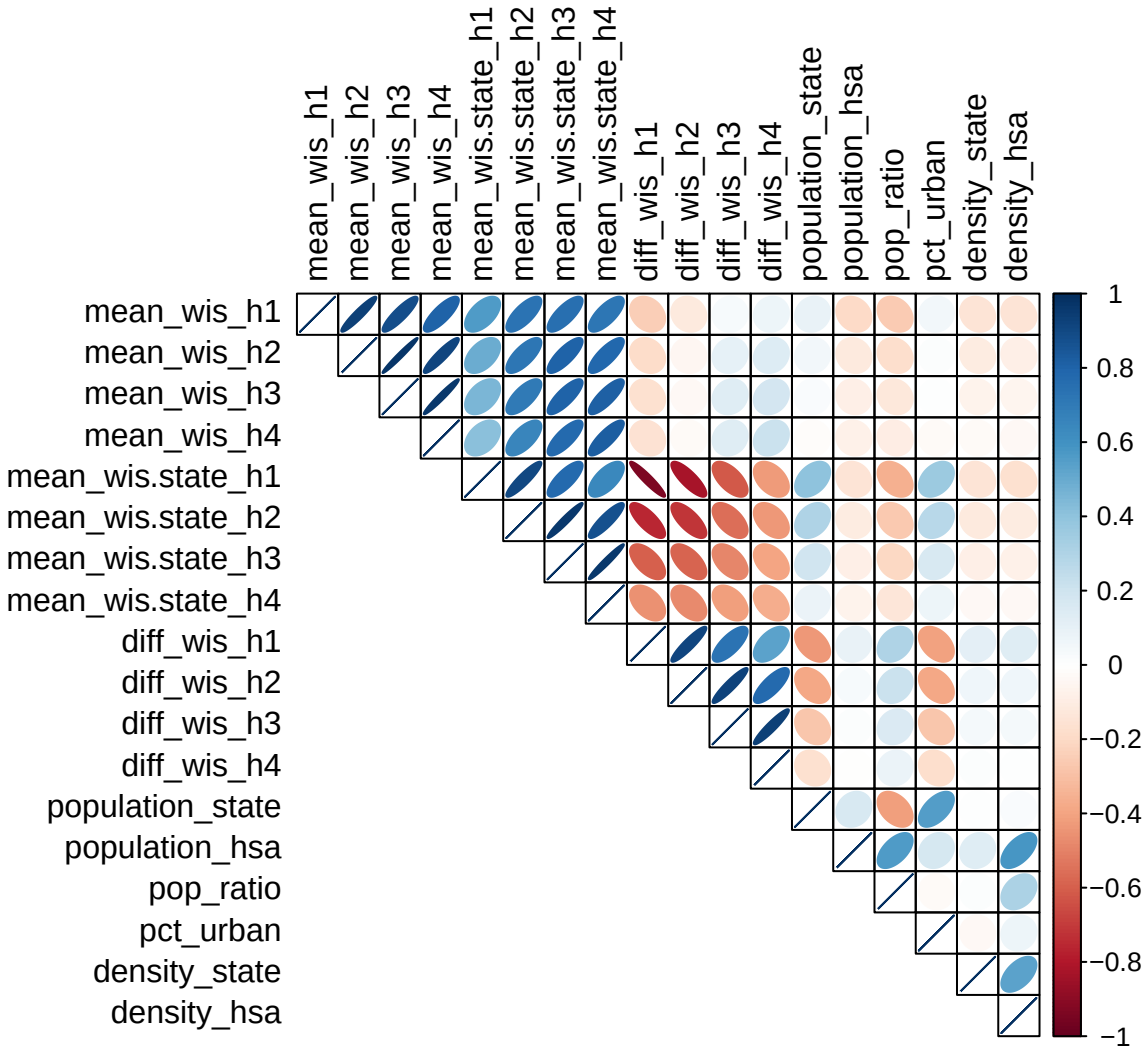
Introduction

This week, I plan to use the methods I mentioned last week to estimate the WIS differences. I'll process and prepare the data and then apply several models to try and see which work best.

Data processing

Load and preprocess data (borrowed from other file)

```
df <- readr::read_csv("GBQR_diff_wis.csv", show_col_types = FALSE)

df_season <- df %>%
  filter(season == "2023/24")

cor_matrix = cor(df[, 4:ncol(df_season)])
# corrplot.mixed(cor_matrix, tl.col = "black", tl.pos = "lt", addgrid.col = TRUE, upper="color", lower=
corrplot(cor_matrix, tl.col = "black", tl.pos = "lt", addgrid.col = TRUE, type = "upper", method = "ell
```


mean_wis_h1
mean_wis_h2
mean_wis_h3
mean_wis_h4
mean_wis.state_h1
mean_wis.state_h2
mean_wis.state_h3
mean_wis.state_h4
diff_wis_h1
diff_wis_h2
diff_wis_h3
diff_wis_h4
population_state
population_hsa
pop_ratio
pct_urban
density_state
density_hsa
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

Support Vector Regression (SVR)

Theoretical background

Implementation

Bayesian Additive Regression Trees

Theoretical background

Implementation

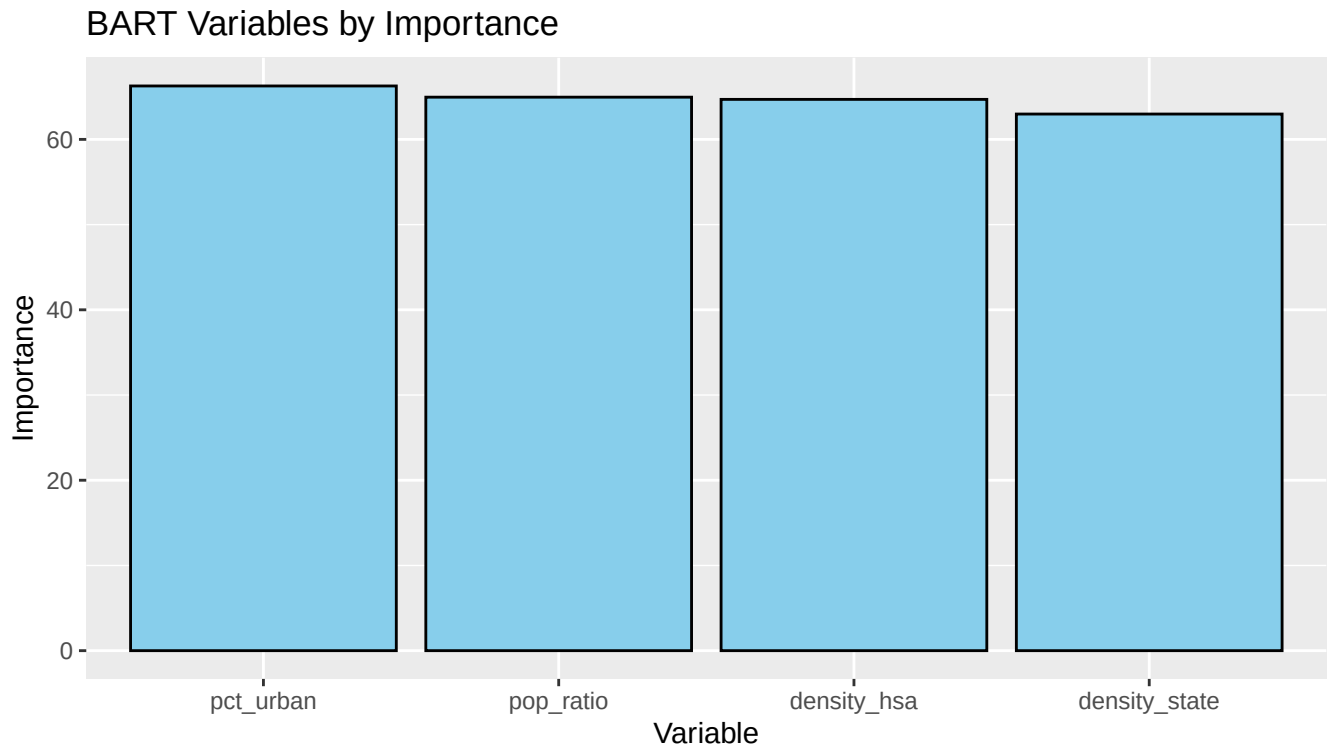
```
library(dbarts)
```

```
##
## Attaching package: 'dbarts'

## The following object is masked from 'package:tidyr':
##
##     extract
```

```
bart_fit <- bart(
  x.train = df_season[, c("pop_ratio","pct_urban",
                          "density_state","density_hsa")],
  y.train = df_season$diff_wis_h3,
  verbose = FALSE, keeptrees = TRUE
)
```

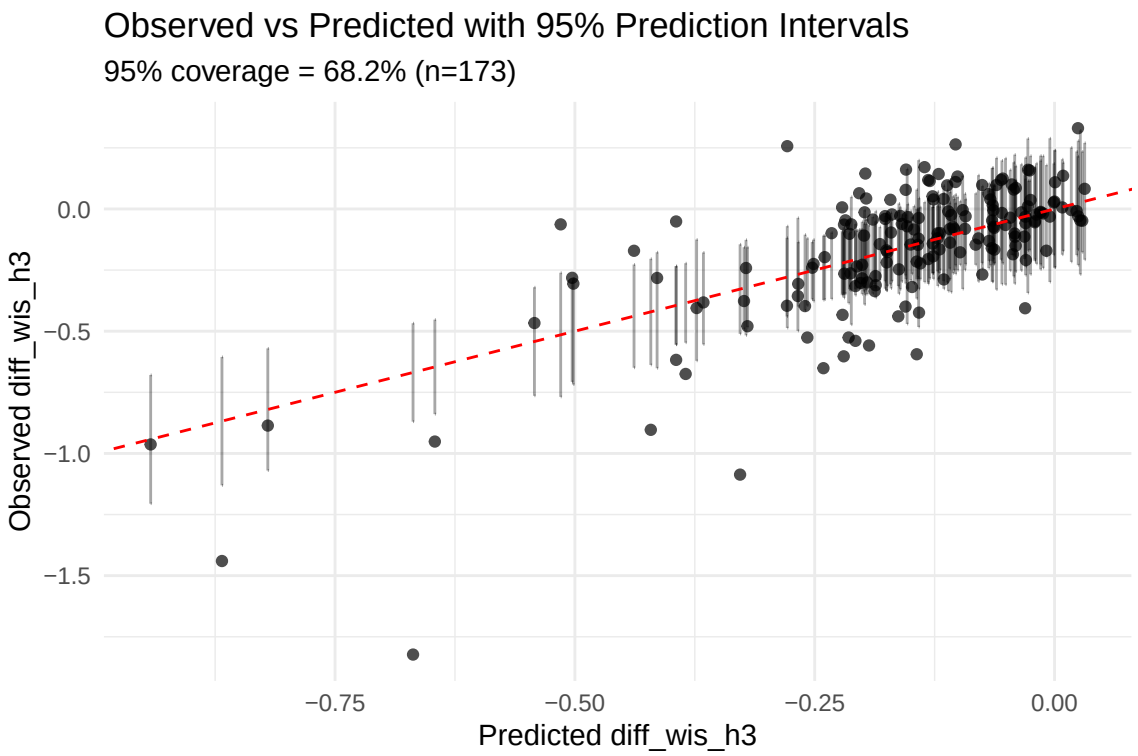
Feature Importance

```
bart_vi = as.data.frame(bart_fit$varcount) %>%
  summarise(across(everything(), ~ mean(.x, na.rm = TRUE))) %>%
  pivot_longer(cols = everything(),
               names_to = "variable",
               values_to = "avg")


ggplot(bart_vi) + geom_col(aes(x = reorder(variable, -avg), y = avg), fill='skyblue', col='black') + lab
```




```
pred = predict(bart_fit, newdata = df_season)
bart_predictions = as.data.frame(t(apply(pred, 2, quantile, probs = c(0.025, 0.975))))
colnames(bart_predictions) = c("ci_low", "ci_high")

bart_predictions$estimate = (bart_predictions$ci_low + bart_predictions$ci_high) / 2
bart_predictions$observed = df_season$diff_wis_h3

coverage_95 = mean(
  bart_predictions$observed >= bart_predictions$ci_low &
  bart_predictions$observed <= bart_predictions$ci_high,
  na.rm = TRUE
)
```

```
ggplot(bart_predictions, aes(x = estimate, y = observed)) +
  geom_errorbar(aes(ymin = ci_low, ymax = ci_high), width = 0, alpha = 0.35) +
  geom_point(alpha = 0.7) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  labs(x = "Predicted diff_wis_h3", y = "Observed diff_wis_h3",
       title = "Observed vs Predicted with 95% Prediction Intervals",
       subtitle = sprintf("95%% coverage = %.1f%% (n=%d)", 100*coverage_95, nrow(bart_predictions))) +
  theme_minimal()
```



Elastic Net / LASSO / Ridge

Theoretical background

Implementation

Conclusion