However, when making predictions on individuals, a biased model may be preferable; biased models may be more accurate, or make predictions with lower relative error than an unbiased model. For example, tree-based ensemble models tend to be highly accurate, and are often the modelling approach of choice for many machine learning applications. In this note, we will show that tree-based models are biased, or uncalibrated. This means they may not always represent the best bias/variance trade-off.

**Example: Predicting income**

We’ll continue the example from the previous post: predicting income from demographic variables (sex, age, employment, education). The data is from the 2016 US Census American Community Survay (ACS) Public Use Microdata Sample (PUMS) for our example. More information about the data can be found [here](https://github.com/WinVector/PDSwR2/tree/master/PUMS). First, we’ll get the training and test data, and show how the expected income varies along different groupings (by sex, by employment, and by education):

library(zeallot)

library(wrapr)

location <- "https://github.com/WinVector/PDSwR2/raw/master/PUMS/incomedata.rds"

incomedata <- readRDS(url(location))

c(test, train) %<-% split(incomedata, incomedata$gp)

# A convenience function to calculate and display

# the conditional expected incomes

show\_conditional\_means <- function(d, outcome = "income") {

cols <- qc(sex, employment, education)

lapply(

cols := cols,

function(colname) {

aggregate(d[, outcome, drop = FALSE],

d[, colname, drop = FALSE],

FUN = mean)

})

}

display\_tables <- function(tlist) {

for(vi in tlist) {

print(knitr::kable(vi))

}

}

display\_tables(

show\_conditional\_means(train))

| **sex** | **income** |
| --- | --- |
| Male | 55755.51 |
| Female | 47718.52 |

| **employment** | **income** |
| --- | --- |
| Employee of a private for profit | 51620.39 |
| Federal government employee | 64250.09 |
| Local government employee | 54740.93 |
| Private not-for-profit employee | 53106.41 |
| Self employed incorporated | 66100.07 |
| Self employed not incorporated | 41346.47 |
| State government employee | 53977.20 |

| **education** | **income** |
| --- | --- |
| no high school diploma | 31883.18 |
| Regular high school diploma | 38052.13 |
| GED or alternative credential | 37273.30 |
| some college credit, no degree | 42991.09 |
| Associate’s degree | 47759.61 |
| Bachelor’s degree | 65668.51 |
| Master’s degree | 79225.87 |
| Professional degree | 97772.60 |
| Doctorate degree | 91214.55 |

**Three models**

We’ll fit three models to the data: two tree ensemble models (random forest and gradient boosting), and one (quasi)Poisson model–a calibrated model– for comparison.

library(randomForest)

library(xgboost)

# Quasipoisson model

model\_pincome <- glm(income ~ age+sex+employment+education,

data=train,

family=quasipoisson)

# random forest model

model\_rf\_1stage <- randomForest(income ~ age+sex+employment+education,

data=train)

# gradient boosting model

# build the model.matrix for the training data

train\_mm <- model.matrix(income ~ age+sex+employment+education,

train)

cvobj <- xgb.cv(params = list(objective="reg:linear"),

train\_mm,

label= train$income,

verbose=0,

nfold=5,

nrounds=50)

evallog <- cvobj$evaluation\_log

ntrees <- which.min(evallog$test\_rmse\_mean)

model\_xgb <- xgboost(train\_mm,

label= train$income,

verbose=0, nrounds=ntrees)

#

# make the predictions on training data

#

train <- transform(train,

pred\_pois = predict(model\_pincome,

train, type="response"),

pred\_rf\_raw = predict(model\_rf\_1stage,

newdata=train, type="response"),

pred\_xgb = predict(model\_xgb, train\_mm))

First, we’ll compare the rollups of the predictions to the actual rollups.

outcomecols <- qc(income, pred\_pois, pred\_rf\_raw, pred\_xgb)

rollups <-show\_conditional\_means(train, outcomecols)

display\_tables(rollups)

| **sex** | **income** | **pred\_pois** | **pred\_rf\_raw** | **pred\_xgb** |
| --- | --- | --- | --- | --- |
| Male | 55755.51 | 55755.51 | 55261.04 | 54203.70 |
| Female | 47718.52 | 47718.52 | 48405.71 | 47326.59 |

| **employment** | **income** | **pred\_pois** | **pred\_rf\_raw** | **pred\_xgb** |
| --- | --- | --- | --- | --- |
| Employee of a private for profit | 51620.39 | 51620.39 | 51276.95 | 50294.95 |
| Federal government employee | 64250.09 | 64250.09 | 61623.87 | 60596.12 |
| Local government employee | 54740.93 | 54740.93 | 55464.36 | 54121.91 |
| Private not-for-profit employee | 53106.41 | 53106.41 | 54135.75 | 53417.86 |
| Self employed incorporated | 66100.07 | 66100.07 | 63840.91 | 63391.52 |
| Self employed not incorporated | 41346.47 | 41346.47 | 46257.98 | 42578.69 |
| State government employee | 53977.20 | 53977.20 | 55530.86 | 54752.98 |

| **education** | **income** | **pred\_pois** | **pred\_rf\_raw** | **pred\_xgb** |
| --- | --- | --- | --- | --- |
| no high school diploma | 31883.18 | 31883.18 | 40599.88 | 40287.54 |
| Regular high school diploma | 38052.13 | 38052.13 | 41864.18 | 36245.78 |
| GED or alternative credential | 37273.30 | 37273.30 | 42316.76 | 37654.63 |
| some college credit, no degree | 42991.09 | 42991.09 | 44303.14 | 41259.59 |
| Associate’s degree | 47759.61 | 47759.61 | 46831.13 | 44995.83 |
| Bachelor’s degree | 65668.51 | 65668.51 | 64131.61 | 64043.09 |
| Master’s degree | 79225.87 | 79225.87 | 70762.24 | 77177.23 |
| Professional degree | 97772.60 | 97772.60 | 77940.16 | 93507.90 |
| Doctorate degree | 91214.55 | 91214.55 | 76972.02 | 86496.11 |

Note that the rollups of the predictions from the two ensemble models don’t match the true rollups, even on the training data; unlike the Poisson model, the random forest and gradient boosting models are uncalibrated.

**Model performance on holdout data**

Let’s see the performance of the models on test data.

# build the model.matrix for the test data

test\_mm <- model.matrix(income ~ age+sex+employment+education,

test)

test <- transform(test,

pred\_pois = predict(model\_pincome,

test, type="response"),

pred\_rf\_raw = predict(model\_rf\_1stage,

newdata=test, type="response"),

pred\_xgb = predict(model\_xgb, test\_mm))

outcomecols <- qc(income, pred\_pois, pred\_rf\_raw, pred\_xgb)

rollups <-show\_conditional\_means(test, outcomecols)

display\_tables(rollups)

| **sex** | **income** | **pred\_pois** | **pred\_rf\_raw** | **pred\_xgb** |
| --- | --- | --- | --- | --- |
| Male | 55408.95 | 55899.83 | 55210.82 | 54236.94 |
| Female | 46261.99 | 47111.01 | 47950.01 | 46705.38 |

| **employment** | **income** | **pred\_pois** | **pred\_rf\_raw** | **pred\_xgb** |
| --- | --- | --- | --- | --- |
| Employee of a private for profit | 50717.96 | 51362.44 | 51040.56 | 49995.11 |
| Federal government employee | 66268.05 | 64881.32 | 61974.36 | 61574.06 |
| Local government employee | 52565.89 | 54119.83 | 54901.92 | 53703.97 |
| Private not-for-profit employee | 52887.52 | 53259.07 | 53987.90 | 53441.93 |
| Self employed incorporated | 67744.61 | 66096.20 | 63790.77 | 63100.00 |
| Self employed not incorporated | 41417.25 | 41507.17 | 46086.63 | 42296.44 |
| State government employee | 51314.92 | 53973.39 | 55262.11 | 54374.54 |

| **education** | **income** | **pred\_pois** | **pred\_rf\_raw** | **pred\_xgb** |
| --- | --- | --- | --- | --- |
| no high school diploma | 29903.70 | 31783.60 | 40469.94 | 40169.21 |
| Regular high school diploma | 36979.33 | 37746.81 | 41648.80 | 35989.04 |
| GED or alternative credential | 39636.86 | 37177.50 | 42620.37 | 38180.68 |
| some college credit, no degree | 43490.42 | 43270.86 | 44449.98 | 41538.77 |
| Associate’s degree | 48384.19 | 47234.56 | 46309.68 | 44383.58 |
| Bachelor’s degree | 65268.96 | 66141.27 | 64387.44 | 64320.68 |
| Master’s degree | 77180.40 | 79594.17 | 70804.81 | 77491.04 |
| Professional degree | 94976.75 | 99009.56 | 78713.55 | 94974.29 |
| Doctorate degree | 87535.83 | 91742.54 | 76517.41 | 86141.53 |

# see how close the rollups get to ground truth for employment

err\_mag <- function(x, y) {

delta = y-x

sqrt(sum(delta^2))

}

employment <- rollups$employment

lapply(employment[, qc(pred\_pois, pred\_rf\_raw, pred\_xgb)],

function(p) err\_mag(p, employment$income)) %.>%

as.data.frame(.) %.>%

knitr::kable(.)

| **pred\_pois** | **pred\_rf\_raw** | **pred\_xgb** |
| --- | --- | --- |
| 3831.967 | 8844.436 | 7474.311 |

The calibrated Poisson model gives better estimates of the income rollups with respect to employment than either of the ensemble models, despite the fact that all the models have similar root mean squared error when making individual predictions.

# predictions on individuals

rmse = function(x, y) {

sqrt(mean((y-x)^2))

}

lapply(test[, qc(pred\_pois, pred\_rf\_raw, pred\_xgb)],

function(p) rmse(p, test$income)) %.>%

as.data.frame(.) %.>%

knitr::kable(.)

| **pred\_pois** | **pred\_rf\_raw** | **pred\_xgb** |
| --- | --- | --- |
| 31341.14 | 31688.77 | 31299.37 |

**Conclusion**

In this example, the input variables were simply not informative enough, so the tree ensemble models performed equivalently to the Poisson model for predicting income. With more informative (and nonlinear) input variables, one can expect that ensemble models will outperform linear or generalized linear models, in terms of predictions on individuals. However, even these more accurate ensemble models can be biased, so they are not guaranteed to estimate important aggregates (grouped sums or conditional means) correctly.

In the next note, we’ll propose a polishing step on uncalibrated models that mitigates this bias, potentially enabling models that are both highly accurate on individuals, while estimating certain aggregates correctly.