Using Cumulative Data

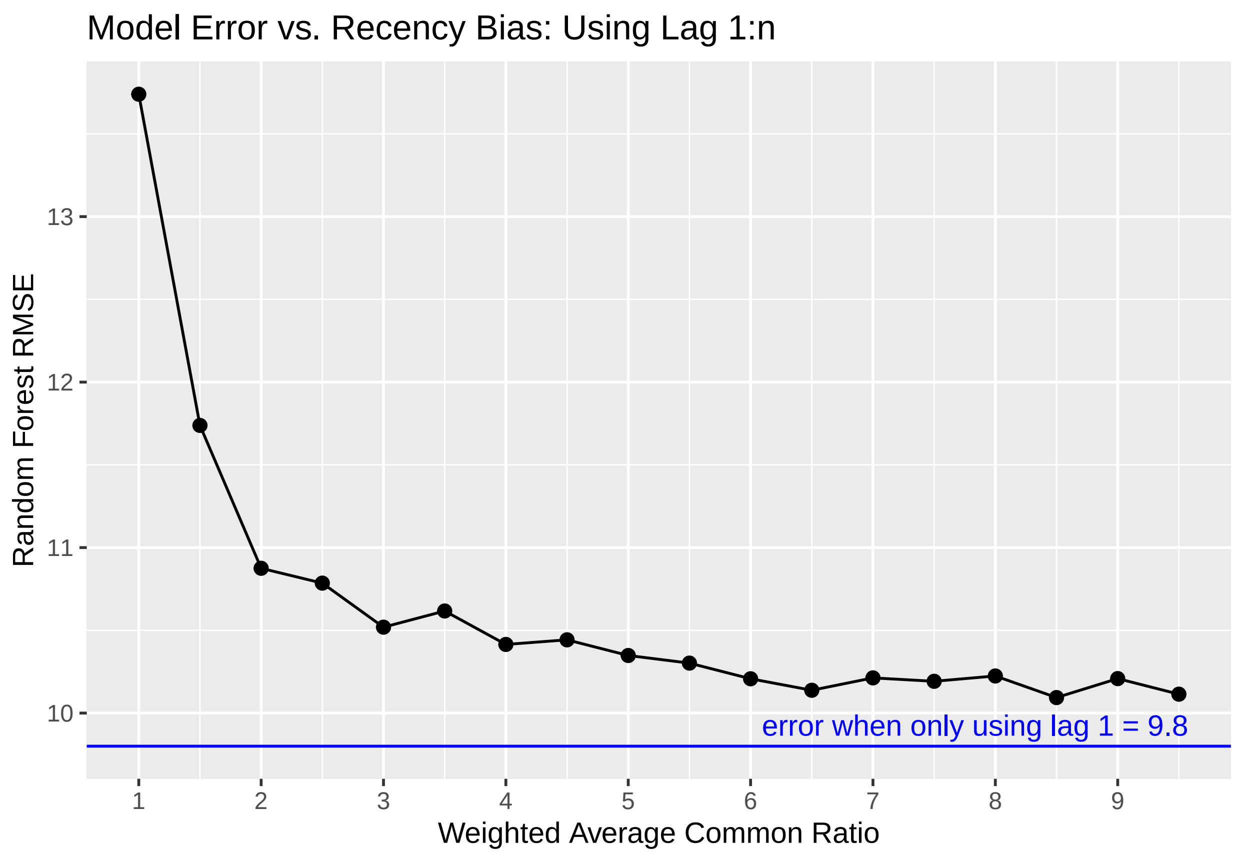
One strategy for improving the predictive power of our models for FHR is incorporating cumulative data. The idea is to all data outside of a given lag period, not just the most recent data. Cumulative data is incorporated by weighted averages. The method for creating weighted averages is a geometric series. The weighted average uses a common ratio, r. r corresponds to the ratio of an observation relative to the one before it. For example, an r of 1.5 means that data from 2024 Q3 has 1.5 times as much weight as data from 2024 Q2. Thus, an r of 1 computes the mean of the data. Several random forests are run on different values of r in order to test the impact of weighing recent data more and more relative to older data. The geometric weighting was chosen because it is easy to implement, especially given that not all suppliers have the same number of data entries. Weights can be divided by r until the oldest data entry is reached and then normalized.

Weighted avg- lag 1:n

```
div_vals = c(1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5)
rmse_vals = c()

for (div in div_vals){

  df_weighted_avg = data.frame()

  for (s in suppliers){
    section = df[df$vlookup_supplier_number == s,]

    nrow = nrow(section)

    if (nrow > 1) {

      new_fhr = section[1, 5]

      new_fhr = new_fhr |> rename(new_fhr = fhr)

      w = 10
      w_sum = 0

      row = as.data.frame(matrix(0, nrow = 1, ncol = 63))
      for (i in 2:nrow) {

        row = (section[i, 4:66] * w) + row

        w_sum = w_sum + w

        w = w / div

      }

      row = row / w_sum

      new_row = cbind(new_fhr, row)

      df_weighted_avg = rbind(df_weighted_avg, new_row)
    }
  }
```

```r
      rf_weighted_avg = randomForest(new_fhr~., data = df_weighted_avg)

      rmse = sqrt(rf_weighted_avg$mse[length(rf_weighted_avg$mse)])

      rmse_vals = c(rmse_vals, rmse)
}

df_div_rmse = data.frame(div_vals, rmse_vals)
```

```r
df_div_rmse |> ggplot(aes(x=div_vals, y = rmse_vals)) + geom_point(size=2) + scale_x_continuous(breaks
```

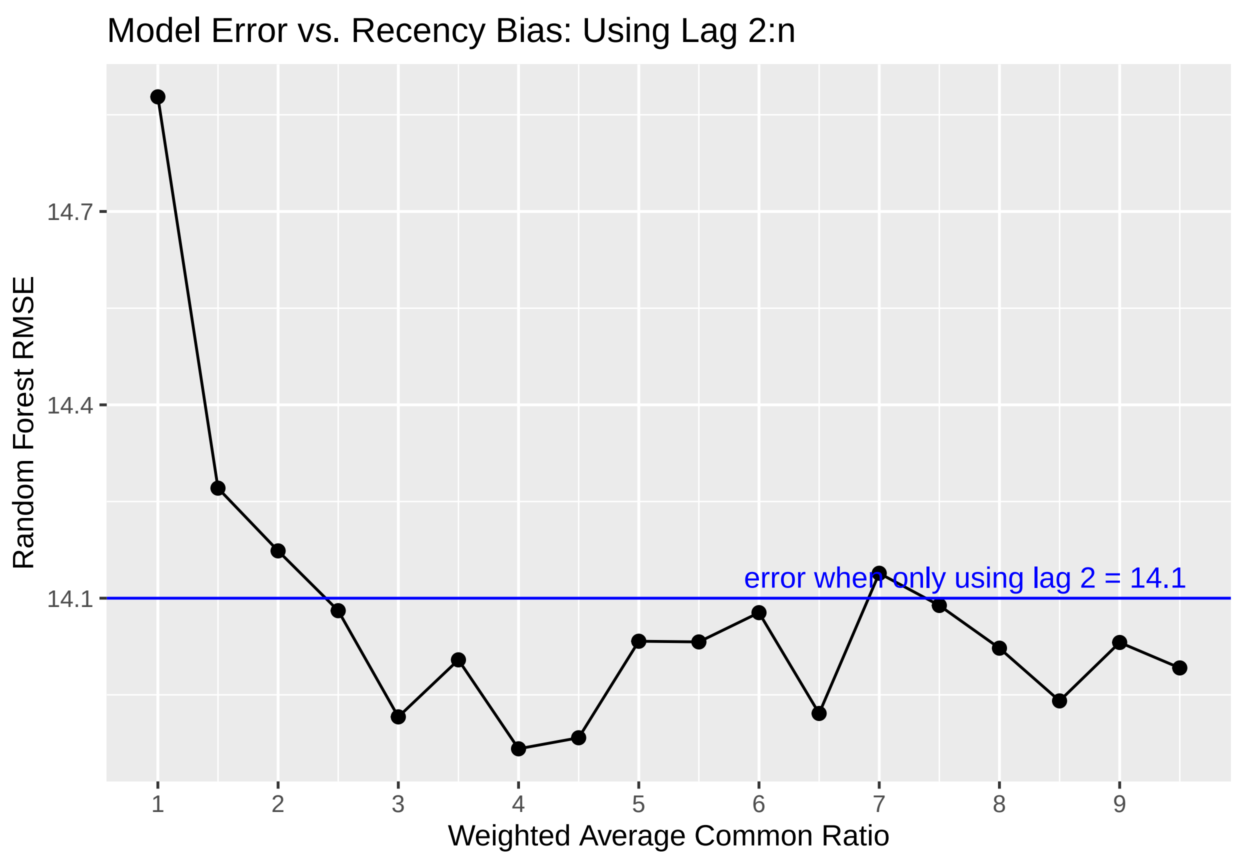


Using cumulative data is not helpful for our lag 1 models. As the weighted average common ratio increases, the random forest testing error decreases. More weight on recent data improves acccuracy.

weighted avg lag 2:n

```r
rmse_vals2 = c()

for (div in div_vals){

  df_weighted_avg2 = data.frame()

  for (s in suppliers){
    section = df[df$vlookup_supplier_number == s,]
    nrow = nrow(section)

    if (nrow > 2) {

      new_fhr = section[1, 5]

      new_fhr = new_fhr |> rename(new_fhr = fhr)

      w = 10
      w_sum = 0

      row = as.data.frame(matrix(0, nrow = 1, ncol = 63))
      for (i in 3:nrow) {

        row = (section[i, 4:66] * w) + row

        w_sum = w_sum + w

        w = w / div

      }

      row = row / w_sum

      new_row = cbind(new_fhr, row)

      df_weighted_avg2 = rbind(df_weighted_avg2, new_row)
    }
  }

      rf_weighted_avg2 = randomForest(new_fhr~., data = df_weighted_avg2)

      rmse = sqrt(rf_weighted_avg2$mse[length(rf_weighted_avg2$mse)])

      rmse_vals2 = c(rmse_vals2, rmse)
}

df_div_rmse2 = data.frame(div_vals, rmse_vals2)
```

```
df_div_rmse2 |> ggplot(aes(x=div_vals, y = rmse_vals2)) + geom_point(size=2) + scale_x_continuous(break
```

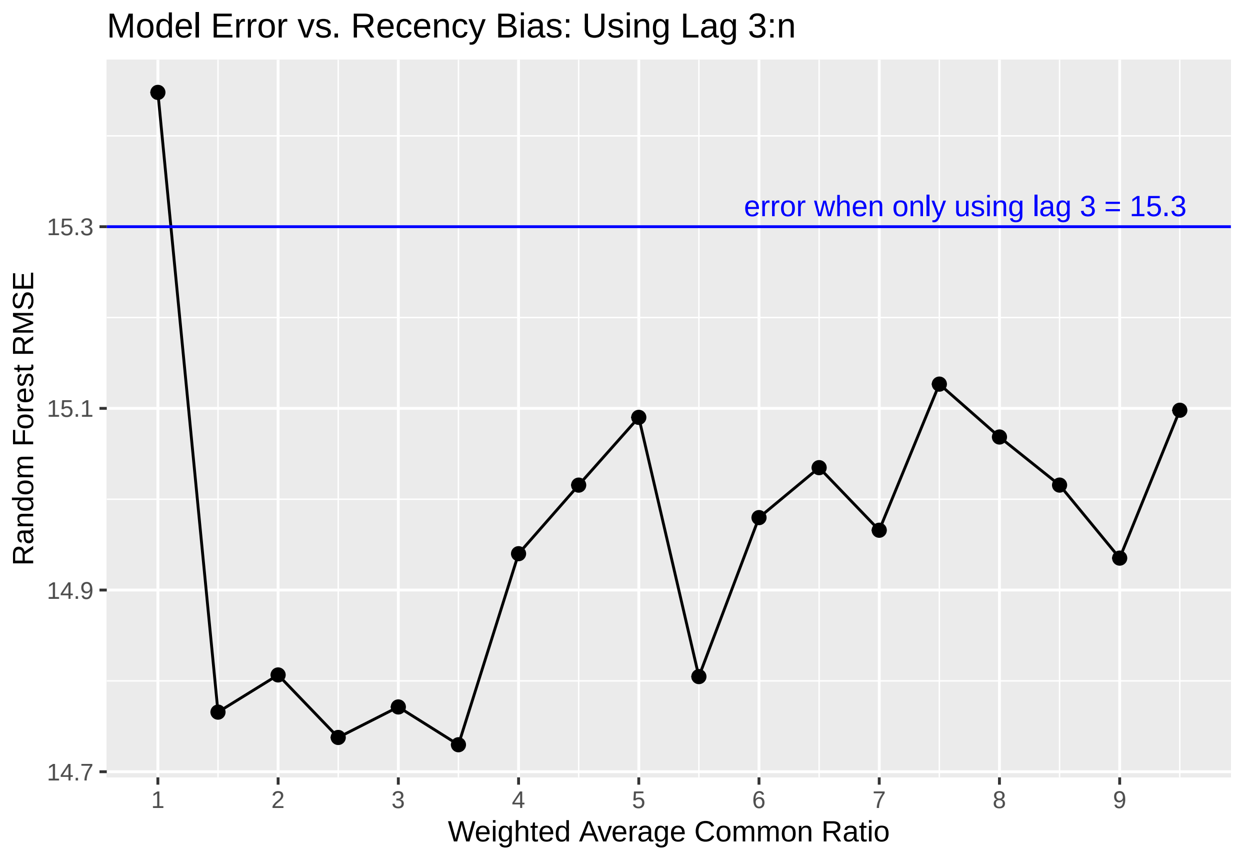


In lag 2:n, using cumulative data leads to slightly lower error than just lag 2. A reasonable way to make predictions about FHR 2 quarters into the future would be a model that weighs recent data 4 times as heavily as the data 1 quarter before it.

weighted avg lag 3:n

```r
rmse_vals3 = c()

for (div in div_vals){

  df_weighted_avg3 = data.frame()

  for (s in suppliers){
    section = df[df$vlookup_supplier_number == s,]

    nrow = nrow(section)

    if (nrow > 3) {

      new_fhr = section[1, 5]

      new_fhr = new_fhr |> rename(new_fhr = fhr)

      w = 10
      w_sum = 0

      row = as.data.frame(matrix(0, nrow = 1, ncol = 63))
      for (i in 4:nrow) {

        row = (section[i, 4:66] * w) + row

        w_sum = w_sum + w

        w = w / div

      }

      row = row / w_sum

      new_row = cbind(new_fhr, row)

      df_weighted_avg3 = rbind(df_weighted_avg3, new_row)
    }
  }

      rf_weighted_avg3 = randomForest(new_fhr~., data = df_weighted_avg3)

      rmse = sqrt(rf_weighted_avg3$mse[length(rf_weighted_avg3$mse)])

      rmse_vals3 = c(rmse_vals3, rmse)
}

df_div_rmse3 = data.frame(div_vals, rmse_vals3)
```

```
df_div_rmse3 |> ggplot(aes(x=div_vals, y = rmse_vals3)) + geom_point(size=2) + scale_x_continuous(break
```



There is significant evidence that using cumulative data improves predictive power looking 3 quarters into the future. It would be appropriate to weigh data from a gien quarter 2.5 times as heavily as the data from the previous quarter when making such predictions.

Conclusions

We have made a Random Forest model that predicts with an RMSE of 9.19 meaning that on average the predictions can be 9.19 points off in predicting the FHR. This proves to be our best performing model. Through visualizing the high coefficients across different variables to see which predicts FHR the best we come to see that the previous FHR, the previous CHR and the previous financial year along with the other fiscal factors prove to be the most important variables in predicting it.

On looking at the predicted FHR scores we can see that the country of origin for both top and worst performing suppliers is in correspondence with our initial analysis.

We also find that when predicting one quarter into the future it is best to only consider the most recently available data.

When predicting mulitple quarters into the future it is best to consider more than just the most recent data and moreover the most recent data should have more weight than older data.