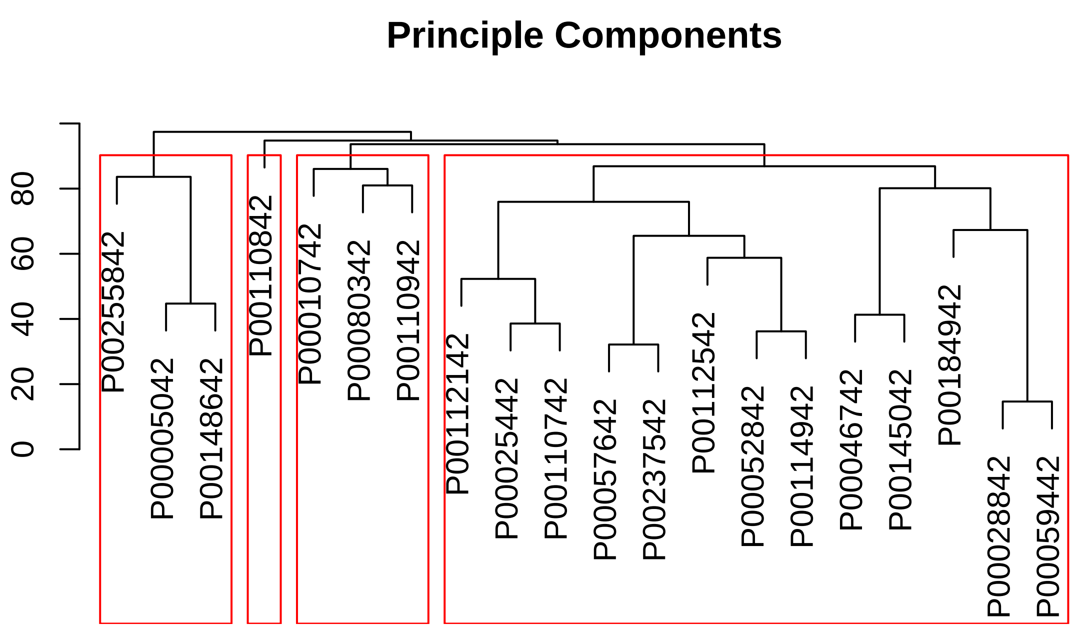


Gradient Boosting

```r
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 3.4.4
```

```
## Loaded gbm 2.1.4
```

```r
set.seed(11)
gbm_split <- initial_split(data.wide, prop = .7)
gbm_train <- training(gbm_split)
gbm_test  <- testing(gbm_split)
```

Perform a grid search which iterates over every combination of hyperparameter values and allows us to assess which combination tends to perform well.

```r
# create hyperparameter grid
hyper_grid <- expand.grid(
  shrinkage = c(.01, .05, .1),
  interaction.depth = c(1, 3, 5),
  n.minobsinnode = c(5, 7, 10),
  bag.fraction = c(0.7, .85, 1),
  optimal_trees = 0,               # a place to dump results
  min_RMSE = 0                     # a place to dump results
)
# randomize data
random_index <- sample(1:nrow(gbm_train), nrow(gbm_train))
random_ames_train <- gbm_train[random_index, ]
# grid search
for( i in 1:nrow(hyper_grid)) {

  # reproducibility
```

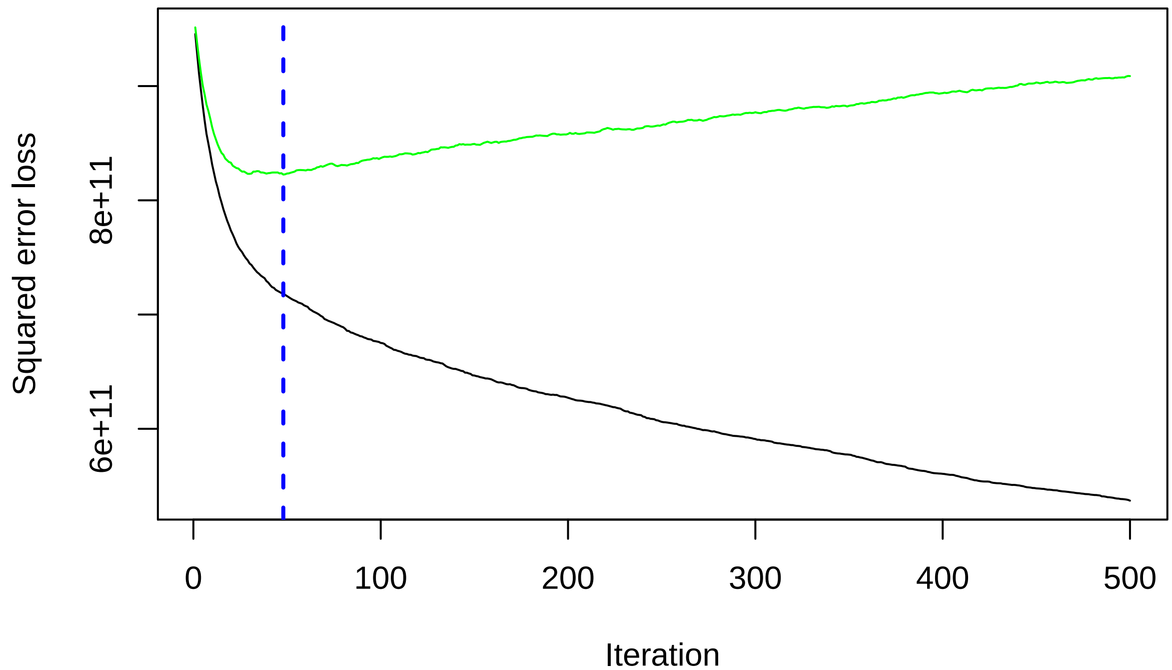
```r
  set.seed(123)

  # train model
  gbm.tune <- gbm(
    formula = purc.total~gender+age+occupation+city_category+stay_years+marital_status,
    distribution = "gaussian",
    data = random_ames_train,
    n.trees = 5000,
    interaction.depth = hyper_grid$interaction.depth[i],
    shrinkage = hyper_grid$shrinkage[i],
    n.minobsinnode = hyper_grid$n.minobsinnode[i],
    bag.fraction = hyper_grid$bag.fraction[i],
    train.fraction = .75,
    n.cores = NULL, # will use all cores by default
    verbose = FALSE
  )

  # add min training error and trees to grid
  hyper_grid$optimal_trees[i] <- which.min(gbm.tune$valid.error)
  hyper_grid$min_RMSE[i] <- sqrt(min(gbm.tune$valid.error))
}

hyper_grid %>%
  dplyr::arrange(min_RMSE) %>%
  head(10)
```

```
##    shrinkage interaction.depth n.minobsinnode bag.fraction optimal_trees
## 1       0.05                 5             10          0.7            68
## 2       0.10                 3              7          1.0            33
## 3       0.10                 3             10          1.0            33
## 4       0.10                 3              5          1.0            26
## 5       0.05                 3             10          1.0            70
## 6       0.05                 3              7          1.0            60
## 7       0.05                 3              5          1.0            65
## 8       0.05                 5              5          0.7            58
## 9       0.01                 3              5          1.0           339
## 10      0.01                 3             10          1.0           359
##    min_RMSE
## 1  823864.6
## 2  823913.3
## 3  823969.5
## 4  824066.3
## 5  824076.9
## 6  824094.2
## 7  824132.1
## 8  824212.1
## 9  824219.1
## 10 824274.3
```

```r
# train a cross validated model using parameters specified above
gbm.fit.final <- gbm(
  purc.total~gender+age+occupation+city_category+stay_years+marital_status,
  data=gbm_test,
  distribution = "gaussian",
```

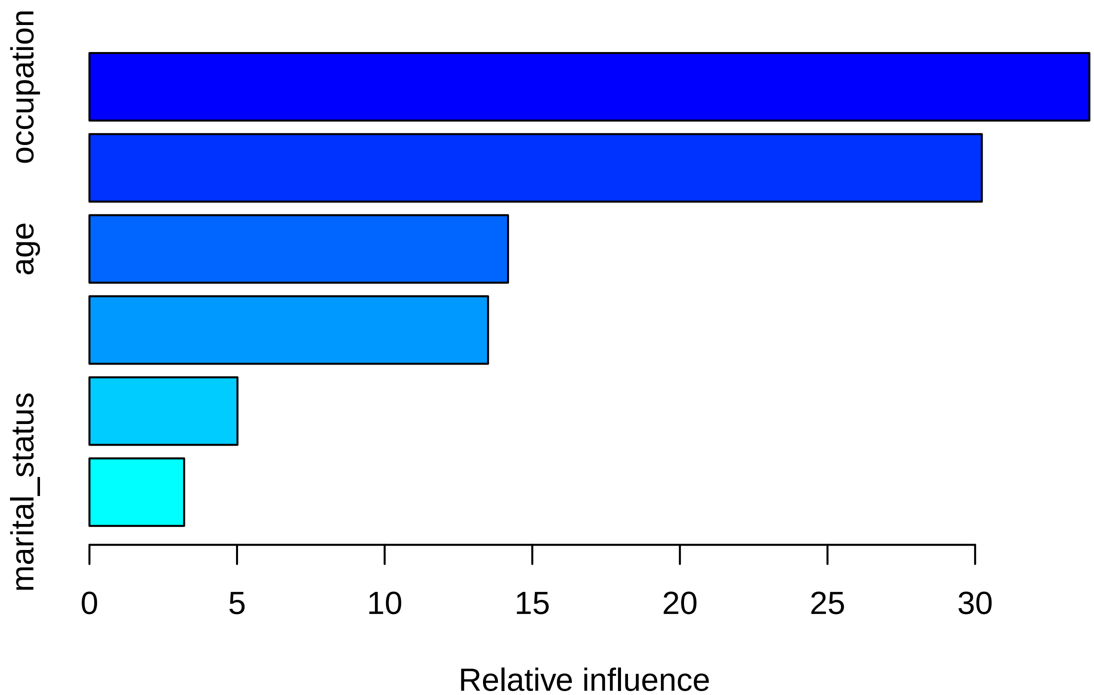
```r
  n.trees = 500,
  interaction.depth = 5, #ensemble a bunch of stumps
  shrinkage = 0.05,
  cv.folds = 5,
  n.minobsinnode = 10,
  bag.fraction=0.70,
  n.cores = NULL, # will use all cores by default
  verbose = FALSE
  )
print(gbm.fit.final)
```

```
## gbm(formula = purc.total ~ gender + age + occupation + city_category +
##     stay_years + marital_status, distribution = "gaussian", data = gbm_test,
##     n.trees = 500, interaction.depth = 5, n.minobsinnode = 10,
##     shrinkage = 0.05, bag.fraction = 0.7, cv.folds = 5, verbose = FALSE,
##     n.cores = NULL)
## A gradient boosted model with gaussian loss function.
## 500 iterations were performed.
```



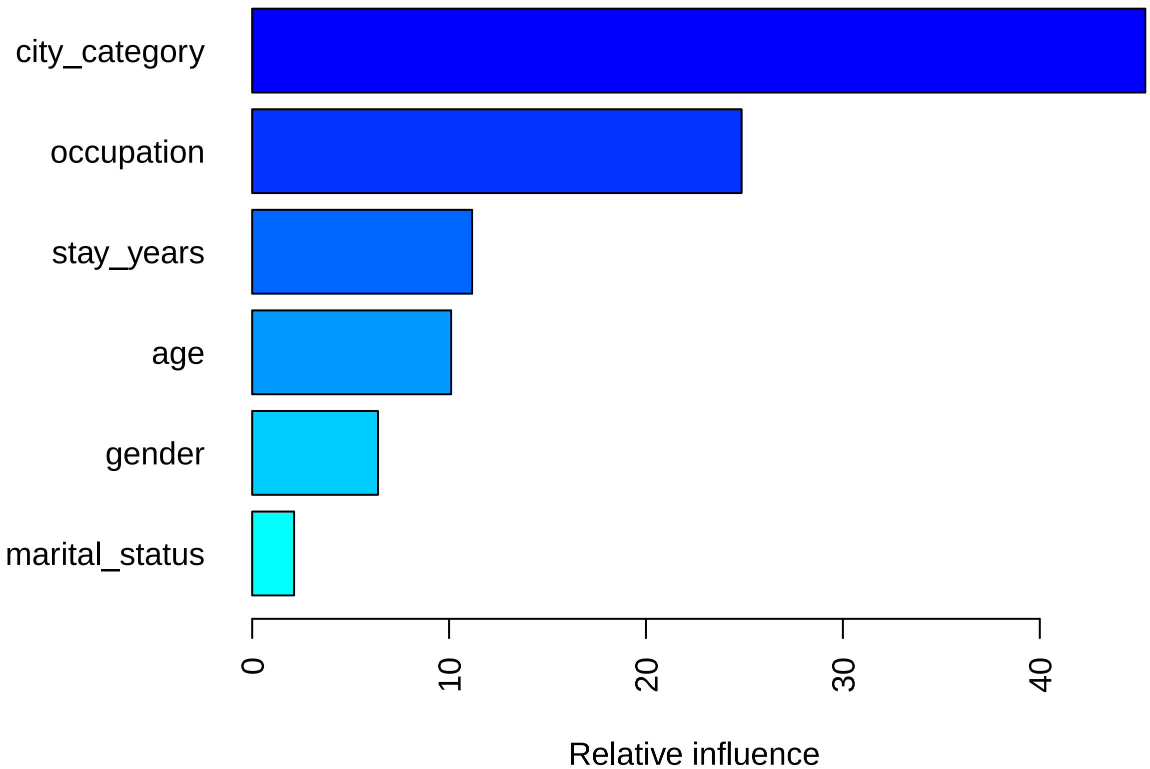

```
## The best cross-validation iteration was 48.
## There were 6 predictors of which 6 had non-zero influence.
```

```r
summary(gbm.fit.final)
```




```
##                             var   rel.inf
## occupation           occupation 33.869892
## city_category     city_category 30.227669
## age                         age 14.178672
## stay_years           stay_years 13.502324
## gender                   gender  5.014235
## marital_status marital_status  3.207209
```

```r
par(mar = c(5, 8, 1, 1))
summary(
  gbm.fit.final,
  cBars = 10,
  #method = relative.influence,
  method=permutation.test.gbm,
  las = 2
  )
```




```
##                var   rel.inf
## 1   city_category 45.356134
## 2      occupation 24.851952
## 3      stay_years 11.176619
## 4             age 10.107433
## 5          gender  6.383976
## 6 marital_status  2.123887
```

```r
pred <- predict(gbm.fit.final, n.trees = gbm.fit.final$n.trees, gbm_test)

# results
caret::RMSE(pred, gbm_test$purc.total)
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2018g.
## 1.0/zoneinfo/America/Detroit'
```

```
## [1] 732861.3
```