eGFR Prediction Using xGboost Regression

Roberts, Haddad, Coady

June 04, 2019

```r
egfr <- read.csv("egfr_clean.csv")[,-1]
set.seed(283749)

train_indicies <- sample(c(1:nrow(egfr)), size=nrow(egfr)*0.5)
valid_indicies <- sample(setdiff(c(1:nrow(egfr)),train_indicies), size=nrow(egfr)*0.25)
test_indicies  <- setdiff(c(1:nrow(egfr)), c(valid_indicies,train_indicies))

nrow(egfr) == length(train_indicies)+length(valid_indicies)+length(test_indicies)
```

```
## [1] TRUE
```

```r
train <- egfr[train_indicies,]
val   <- egfr[valid_indicies,]
test  <- egfr[test_indicies, ]

setDT(train)
setDT(val)
setDT(test)

train_y <- train$score_18
val_y   <- val$score_18
test_y  <- test$score_18

train_X <- train[,-c("score_18")]
val_X   <- val[,-c("score_18")]
test_X  <- test[,-c("score_18")]

dtrain <- xgb.DMatrix(data=as.matrix(train_X),label=train_y)
dval   <- xgb.DMatrix(data=as.matrix(val_X),label=val_y)
```

GET FEATURE IMPORTANCE

```r
params <- list(booster="gbtree",metrics="test_rmse",eta=0.3,gamma=0,max_depth=6,min_child_weight=1,subsa
xgb1 <- xgb.train(data=dtrain,nrounds=100,watchlist=list(train=dtrain),print.every.n=10,early.stop.roun
```

```
## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:50.224701
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:9.180502
## [21] train-rmse:8.725348
## [31] train-rmse:8.483500
## [41] train-rmse:8.289080
## [51] train-rmse:8.061038
## [61] train-rmse:7.954932
## [71] train-rmse:7.790457
## [81] train-rmse:7.639465
## [91] train-rmse:7.509202
## [100]    train-rmse:7.386498
```

```
best_score<-xgb1$best_score

cat("best residual mean squared error", best_score, "\n")
```

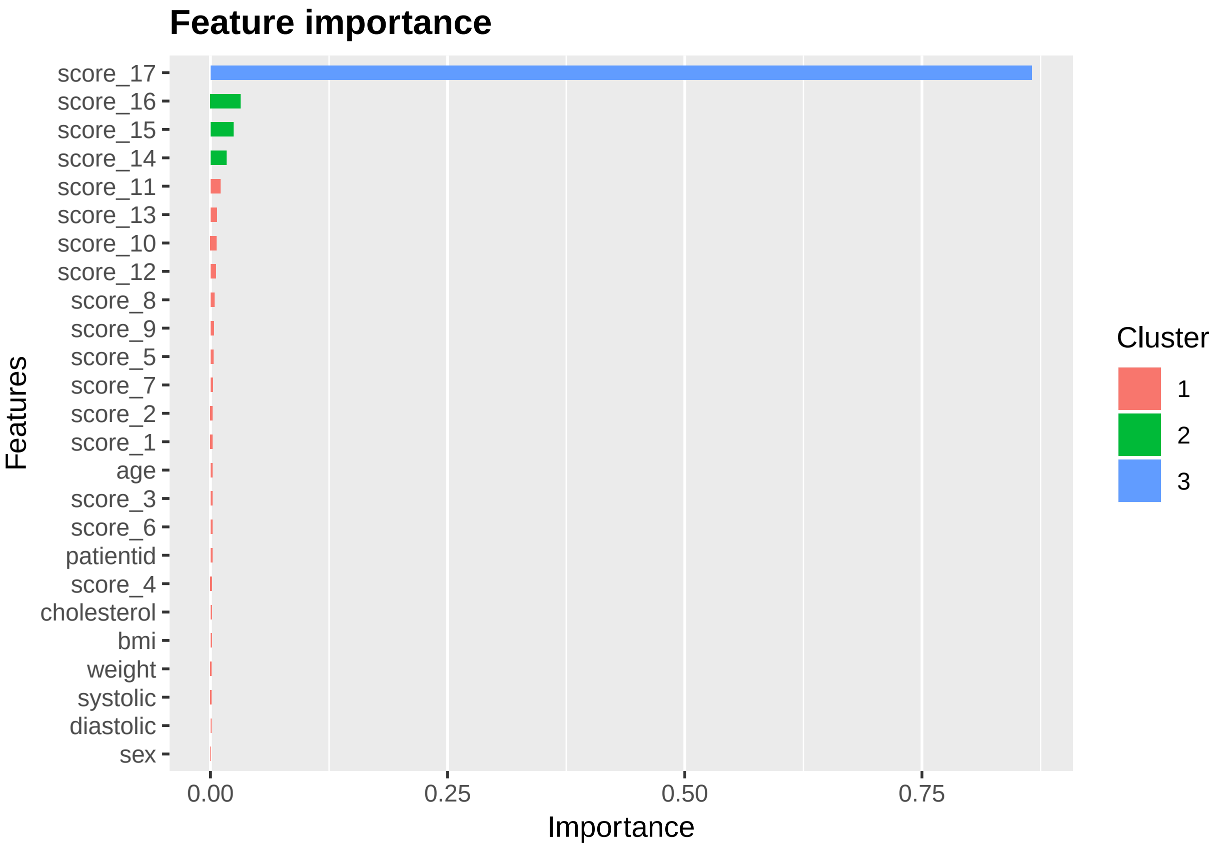
```
## best residual mean squared error 7.386498
```

```
y_pred <- predict(xgb1,as.matrix(val_X))
sse <- sum((val_y - y_pred) ** 2)
tse <- sum((val_y - mean(val_y)) ** 2)
r_squared<-1-(sse/tse)

cat("r-squared", r_squared, "\n")
```

```
## r-squared 0.896325
```

```
important_Df <- xgb.importance(model=xgb1)
```

```
xgb.ggplot.importance(importance_matrix=important_Df)
```



```
#xgb.ggplot.deepness(model = xgb1, which = c("2x1", "max.depth","med.depth", "med.weight"))

#xgb.plot.multi.trees(xgb1)
```

```
write.csv(file="feature_importance.csv", x=important_Df)
```

test reducing model complexity

```
iterations = nrow(important_Df)-3
results <- matrix(ncol=2, nrow=iterations)

for(row in 1:iterations){
  print(row)
  last_col <- nrow(important_Df)-row
  egfr_loop <- egfr[c(c(important_Df[1:last_col,]$Feature), "score_18")]

  train_mc <- egfr_loop[train_indicies,]
  val_mc   <- egfr_loop[valid_indicies,]

  setDT(train_mc)
  setDT(val_mc)

  train_mc_y <- train_mc$score_18
  val_mc_y   <- val_mc$score_18

  train_mc_X <- train_mc[,-c("score_18")]
  val_mc_X   <- val_mc[,-c("score_18")]

  dtrain_mc <- xgb.DMatrix(data=as.matrix(train_mc_X),label=train_mc_y)
  dval_mc   <- xgb.DMatrix(data=as.matrix(val_mc_X),label=val_mc_y)

  params <- list(booster="gbtree",metrics="test_rmse",eta=0.1,gamma=0,max_depth=10,min_child_weight=1,su
  xgb1 <- xgb.train(params=params,data=dtrain_mc,nrounds=700,watchlist=list(train=dtrain_mc),print.every

  y_mc_pred <- predict(xgb1,as.matrix(train_mc_X))
  sse <- sum((train_mc_y - y_mc_pred) ** 2)
  tse <- sum((train_mc_y - mean(train_mc_y)) ** 2)
  r_squared_train<-1-(sse/tse)

  y_mc_pred <- predict(xgb1,as.matrix(val_mc_X))
  sse <- sum((val_mc_y - y_mc_pred) ** 2)
  tse <- sum((val_mc_y - mean(val_mc_y)) ** 2)
  r_squared_val<-1-(sse/tse)

  results[row,] <- c(r_squared_train,r_squared_val)
}
```

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

```
## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").
## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.008385
## Will train until train_rmse hasn't improved in 10 rounds.
##
## Stopping. Best iteration:
## [592]    train-rmse:1.960640
##
## [1] 2

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.008385
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:1.346159
## [1] 3

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.008446
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:1.425338
## [1] 4

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.008446
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:1.425958
## [1] 5

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").
## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.008446
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:1.525575
## [1] 6

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").


## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.008461
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:1.581577
## [1] 7

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").


## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.008537
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:1.730471
## [1] 8

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").


## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.008537
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:1.343933
## [1] 9

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").


## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.008583
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:1.468296
## [1] 10

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.008583
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:1.474059
## [1] 11

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.008575
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:1.790619
## [1] 12

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.008583
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:1.878177
## [1] 13

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").
## [1]  train-rmse:64.008575
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:2.038144
## [1] 14
```

```
## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").
```

```
## [1]  train-rmse:64.008575
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:2.142304
## [1] 15
```

```
## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").
```

```
## [1]  train-rmse:64.008797
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:2.461852
## [1] 16
```

```
## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").
```

```
## [1]  train-rmse:64.009094
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:2.722935
## [1] 17
```

```
## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").
```

```
## [1]  train-rmse:64.009109
## Will train until train_rmse hasn't improved in 10 rounds.
```

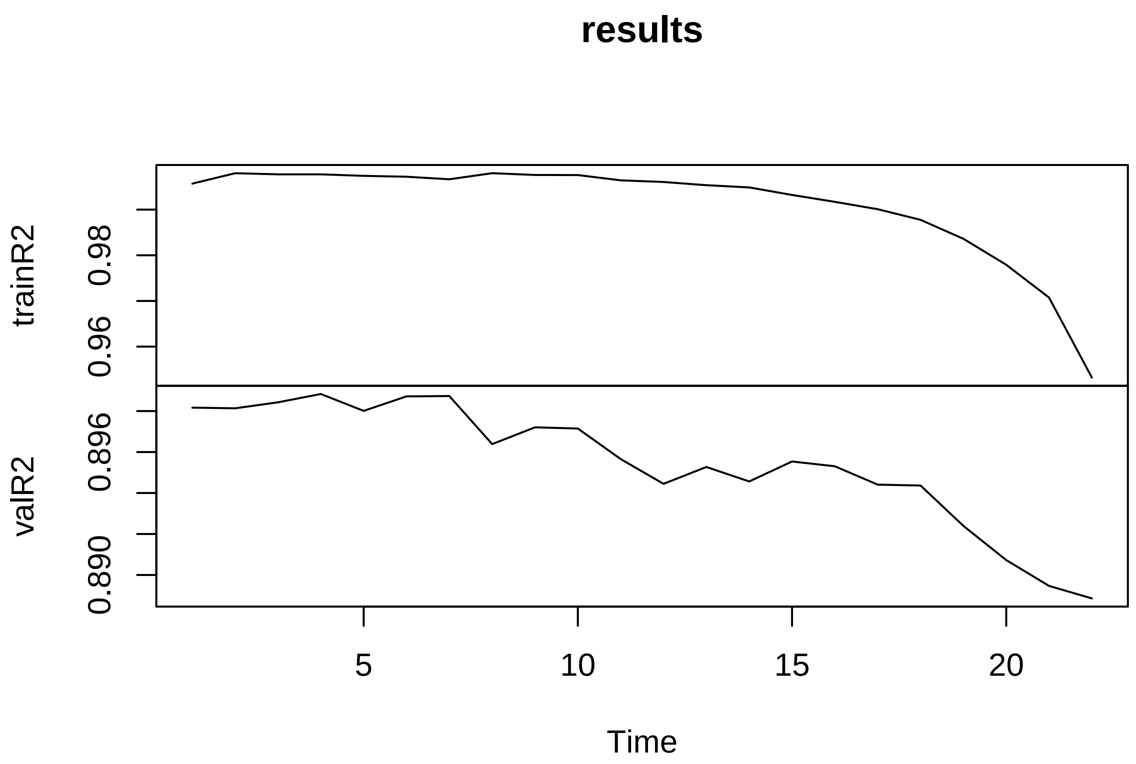
```
##
## [700]    train-rmse:2.975274
## [1] 18

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.009384
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:3.305638
## [1] 19

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.009590
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:3.825246
## [1] 20

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.010223
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:4.439545
## [1] 21

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.010223
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:5.111858
## [1] 22

## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").

## [1]  train-rmse:64.011887
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [700]    train-rmse:6.463403
```

```
results <- data.frame(results)
colnames(results) <- c("trainR2", "valR2")
```

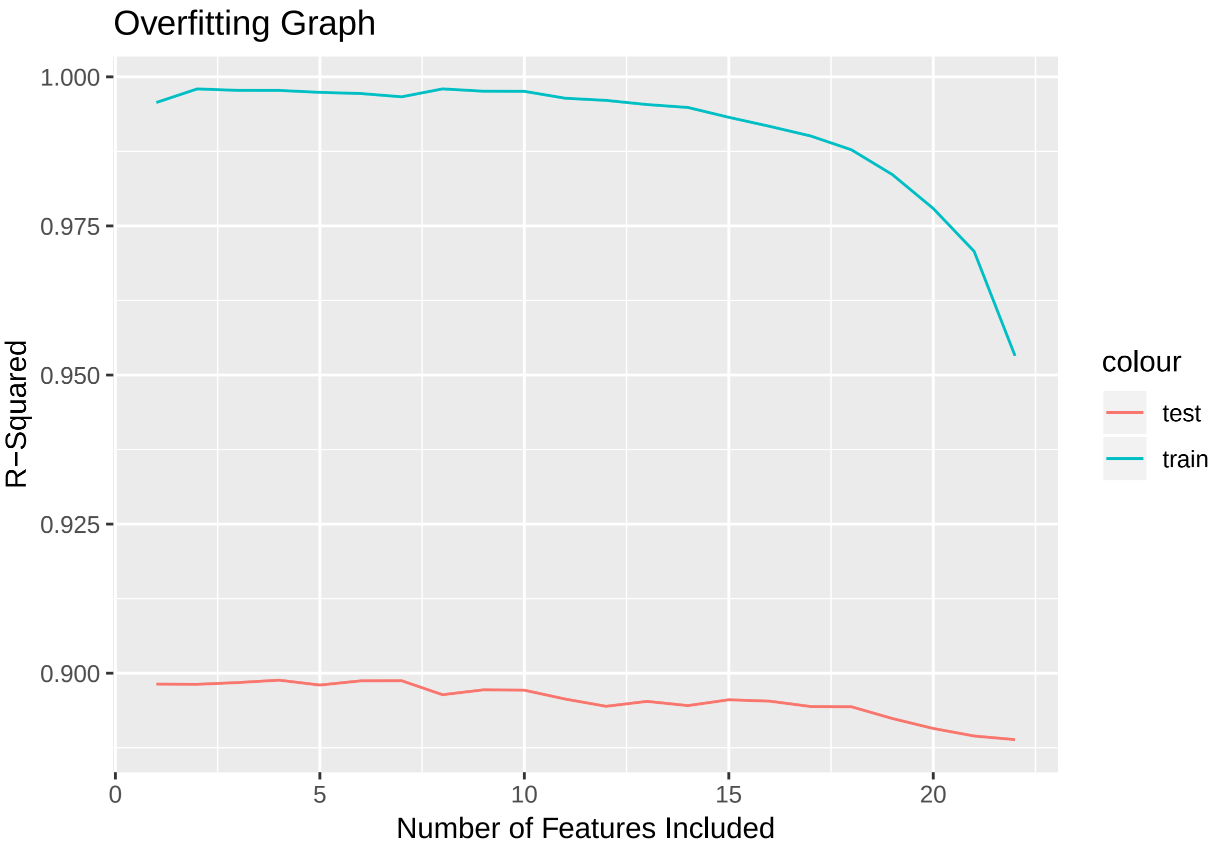
```
results
```

```
##      trainR2     valR2
## 1  0.9956944 0.8981664
## 2  0.9979703 0.8981345
## 3  0.9977245 0.8984267
## 4  0.9977225 0.8988331
## 5  0.9973932 0.8980092
## 6  0.9971983 0.8987169
## 7  0.9966459 0.8987327
## 8  0.9979770 0.8963878
## 9  0.9975853 0.8972059
## 10 0.9975663 0.8971468
## 11 0.9964087 0.8956570
## 12 0.9960489 0.8944481
## 13 0.9953472 0.8952706
## 14 0.9948595 0.8945637
## 15 0.9932116 0.8955387
## 16 0.9916954 0.8953040
## 17 0.9900849 0.8944098
## 18 0.9877609 0.8943664
## 19 0.9836107 0.8923962
## 20 0.9779241 0.8907276
## 21 0.9707315 0.8894662
## 22 0.9532088 0.8888557
```

```
plot.ts(results)
```



```
ggplot(results, aes(seq(1:nrow(results)))) +
  geom_line(aes(y = trainR2, colour = "train")) +
  geom_line(aes(y = valR2, colour = "test")) +
  ggtitle("Overfitting Graph") +
  xlab("Number of Features Included") +
  ylab("R-Squared")
```



```
write.csv(file="feature_selection_results.csv", x=results)
```

```
cat("keep the top", nrow(important_Df) - 11, "most important features")
```

```
## keep the top 14 most important features
```

Best number of features is ... 14

GRID SEARCH

```
egfr <- egfr[c(c(important_Df[1:14,]$Feature), "score_18")]

set.seed(283749)

train_indicies <- sample(c(1:nrow(egfr)), size=nrow(egfr)*0.8)
valid_indicies <- setdiff(c(1:nrow(egfr)), c(train_indicies))

nrow(egfr) == length(train_indicies)+length(valid_indicies)
```

```
## [1] TRUE
```

```
train <- egfr[train_indicies,]
val   <- egfr[valid_indicies,]

# prepare training scheme
control <- trainControl(method="repeatedcv", number=3, repeats=5)

# design the parameter tuning grid
```

```r
grid <- expand.grid(eta=c(0.1,0.3,0.5), max_depth=c(3,7,11),
                    colsample_bytree = seq(0.5, 0.9, length.out = 5),
                    min_child_weight=1,gamma=0,subsample=1,nrounds=100)

# train the model
model <- train(score_18~., data=train, method="xgbTree", trControl=control, tuneGrid=grid)
# summarize the model
print(model)
```

```
## eXtreme Gradient Boosting
##
## 81164 samples
##    14 predictor
##
## No pre-processing
## Resampling: Cross-Validated (3 fold, repeated 5 times)
## Summary of sample sizes: 54108, 54110, 54110, 54109, 54108, 54111, ...
## Resampling results across tuning parameters:
##
##   eta  max_depth  colsample_bytree  RMSE       Rsquared   MAE
##   0.1   3         0.5                9.936163  0.8896635  7.650885
##   0.1   3         0.6                9.891961  0.8906358  7.653404
##   0.1   3         0.7                9.878848  0.8909262  7.644083
##   0.1   3         0.8                9.853880  0.8914795  7.636509
##   0.1   3         0.9                9.840631  0.8917762  7.626872
##   0.1   7         0.5                9.652411  0.8958685  7.105360
##   0.1   7         0.6                9.636997  0.8961902  7.077778
##   0.1   7         0.7                9.610478  0.8967637  7.056566
##   0.1   7         0.8                9.595987  0.8970691  7.007428
##   0.1   7         0.9                9.592822  0.8971350  6.993612
##   0.1  11         0.5                9.664521  0.8955852  6.953700
##   0.1  11         0.6                9.658130  0.8957395  6.926546
##   0.1  11         0.7                9.605797  0.8968620  6.885766
##   0.1  11         0.8                9.585395  0.8972996  6.843488
##   0.1  11         0.9                9.595196  0.8970863  6.844686
##   0.3   3         0.5                9.877149  0.8909571  7.467714
##   0.3   3         0.6                9.846997  0.8916165  7.445610
##   0.3   3         0.7                9.827310  0.8920574  7.441357
##   0.3   3         0.8                9.826703  0.8920760  7.401284
##   0.3   3         0.9                9.830074  0.8919953  7.397709
##   0.3   7         0.5                9.864006  0.8912549  7.103496
##   0.3   7         0.6                9.839772  0.8917957  7.097672
##   0.3   7         0.7                9.846944  0.8916534  7.092819
##   0.3   7         0.8                9.815562  0.8923455  7.044774
##   0.3   7         0.9                9.803341  0.8926060  7.031166
##   0.3  11         0.5               10.119146  0.8856457  7.207232
##   0.3  11         0.6               10.094095  0.8862136  7.176782
##   0.3  11         0.7               10.075974  0.8866317  7.148413
##   0.3  11         0.8               10.088926  0.8863769  7.134961
##   0.3  11         0.9               10.066177  0.8868971  7.105350
##   0.5   3         0.5                9.925340  0.8899143  7.404355
##   0.5   3         0.6                9.902719  0.8904228  7.391935
##   0.5   3         0.7                9.908620  0.8902853  7.368965
##   0.5   3         0.8                9.880423  0.8908999  7.340868
##   0.5   3         0.9                9.869748  0.8911455  7.329796
##   0.5   7         0.5               10.212800  0.8836201  7.296087
##   0.5   7         0.6               10.306980  0.8815340  7.347618
##   0.5   7         0.7               10.231604  0.8832634  7.264776
##   0.5   7         0.8               10.233245  0.8832187  7.252469
##   0.5   7         0.9               10.200662  0.8839642  7.237535
##   0.5  11         0.5               10.769752  0.8710683  7.644010
##   0.5  11         0.6               10.732712  0.8719881  7.623433
##   0.5  11         0.7               10.718366  0.8723089  7.614080
##   0.5  11         0.8               10.714160  0.8725090  7.576604
##   0.5  11         0.9               10.734832  0.8719989  7.580524
##
## Tuning parameter 'nrounds' was held constant at a value of 100
##
## Tuning parameter 'min_child_weight' was held constant at a value of
##  1
## Tuning parameter 'subsample' was held constant at a value of 1
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were nrounds = 100, max_depth =
##  11, eta = 0.1, gamma = 0, colsample_bytree = 0.8, min_child_weight =
##  1 and subsample = 1.
```

```
model$results
```

```
##    eta max_depth gamma colsample_bytree min_child_weight subsample nrounds
## 1  0.1         3     0              0.5                1         1     100
## 2  0.1         3     0              0.6                1         1     100
## 3  0.1         3     0              0.7                1         1     100
## 4  0.1         3     0              0.8                1         1     100
## 5  0.1         3     0              0.9                1         1     100
## 16 0.3         3     0              0.5                1         1     100
## 17 0.3         3     0              0.6                1         1     100
## 18 0.3         3     0              0.7                1         1     100
## 19 0.3         3     0              0.8                1         1     100
## 20 0.3         3     0              0.9                1         1     100
## 31 0.5         3     0              0.5                1         1     100
## 32 0.5         3     0              0.6                1         1     100
## 33 0.5         3     0              0.7                1         1     100
## 34 0.5         3     0              0.8                1         1     100
## 35 0.5         3     0              0.9                1         1     100
## 6  0.1         7     0              0.5                1         1     100
## 7  0.1         7     0              0.6                1         1     100
## 8  0.1         7     0              0.7                1         1     100
## 9  0.1         7     0              0.8                1         1     100
## 10 0.1         7     0              0.9                1         1     100
## 21 0.3         7     0              0.5                1         1     100
## 22 0.3         7     0              0.6                1         1     100
## 23 0.3         7     0              0.7                1         1     100
## 24 0.3         7     0              0.8                1         1     100
## 25 0.3         7     0              0.9                1         1     100
## 36 0.5         7     0              0.5                1         1     100
## 37 0.5         7     0              0.6                1         1     100
## 38 0.5         7     0              0.7                1         1     100
## 39 0.5         7     0              0.8                1         1     100
## 40 0.5         7     0              0.9                1         1     100
## 11 0.1        11     0              0.5                1         1     100
## 12 0.1        11     0              0.6                1         1     100
## 13 0.1        11     0              0.7                1         1     100
## 14 0.1        11     0              0.8                1         1     100
## 15 0.1        11     0              0.9                1         1     100
## 26 0.3        11     0              0.5                1         1     100
## 27 0.3        11     0              0.6                1         1     100
## 28 0.3        11     0              0.7                1         1     100
## 29 0.3        11     0              0.8                1         1     100
## 30 0.3        11     0              0.9                1         1     100
## 41 0.5        11     0              0.5                1         1     100
## 42 0.5        11     0              0.6                1         1     100
## 43 0.5        11     0              0.7                1         1     100
## 44 0.5        11     0              0.8                1         1     100
## 45 0.5        11     0              0.9                1         1     100
##         RMSE  Rsquared      MAE     RMSESD  RsquaredSD      MAESD
## 1   9.936163 0.8896635 7.650885 0.09500481 0.001556527 0.04326019
## 2   9.891961 0.8906358 7.653404 0.10792603 0.001979878 0.02485967
## 3   9.878848 0.8909262 7.644083 0.11053390 0.001966474 0.02422295
## 4   9.853880 0.8914795 7.636509 0.08297409 0.001516941 0.02583825
## 5   9.840631 0.8917762 7.626872 0.08175323 0.001488526 0.02517770
## 16  9.877149 0.8909571 7.467714 0.11805771 0.002343092 0.04514854
## 17  9.846997 0.8916165 7.445610 0.12454821 0.002391611 0.03937027
## 18  9.827310 0.8920574 7.441357 0.09691878 0.001698696 0.03684157
## 19  9.826703 0.8920760 7.401284 0.11521322 0.002137005 0.02887840
## 20  9.830074 0.8919953 7.397709 0.10792240 0.001848523 0.03190022
## 31  9.925340 0.8899143 7.404355 0.15869701 0.002984625 0.04052406
## 32  9.902719 0.8904228 7.391935 0.14539740 0.002664603 0.04601307
## 33  9.908620 0.8902853 7.368965 0.14202100 0.002705356 0.03771541
## 34  9.880423 0.8908999 7.340868 0.13237713 0.002533475 0.03619940
## 35  9.869748 0.8911455 7.329796 0.10651688 0.001903904 0.03744504
## 6   9.652411 0.8958685 7.105360 0.09608543 0.001499159 0.03688581
## 7   9.636997 0.8961902 7.077778 0.12355102 0.002331703 0.04859490
## 8   9.610478 0.8967637 7.056566 0.09721016 0.001732635 0.04483005
## 9   9.595987 0.8970691 7.007428 0.10583881 0.001885858 0.02524978
## 10  9.592822 0.8971350 6.993612 0.10381846 0.001816226 0.03105757
## 21  9.864006 0.8912549 7.103496 0.14067085 0.002629293 0.04661736
## 22  9.839772 0.8917957 7.097672 0.13092617 0.002451828 0.05556730
## 23  9.846944 0.8916534 7.092819 0.11210967 0.001984877 0.03228806
## 24  9.815562 0.8923455 7.044774 0.10731319 0.002004245 0.03735494
## 25  9.803341 0.8926060 7.031166 0.10727479 0.002044242 0.04026204
## 36 10.212800 0.8836201 7.296087 0.14401852 0.003062605 0.07692476
## 37 10.306980 0.8815340 7.347618 0.15999722 0.003407795 0.08686962
## 38 10.231604 0.8832634 7.264776 0.14461612 0.002840472 0.06857960
## 39 10.233245 0.8832187 7.252469 0.16472388 0.003221766 0.07269217
## 40 10.200662 0.8839642 7.237535 0.09147273 0.001710181 0.04663948
## 11  9.664521 0.8955852 6.953700 0.15564112 0.003245240 0.06083859
## 12  9.658130 0.8957395 6.926546 0.10665617 0.002025206 0.04567577
## 13  9.605797 0.8968620 6.885766 0.11006031 0.002023086 0.04290323
## 14  9.585395 0.8972996 6.843488 0.09039637 0.001441286 0.02467283
## 15  9.595196 0.8970863 6.844686 0.08485302 0.001435583 0.03144543
## 26 10.119146 0.8856457 7.207232 0.15702551 0.003129534 0.06647082
## 27 10.094095 0.8862136 7.176782 0.11864792 0.002427850 0.05295707
## 28 10.075974 0.8866317 7.148413 0.12120415 0.002460617 0.05303563
## 29 10.088926 0.8863769 7.134961 0.11666562 0.002295923 0.05368725
## 30 10.066177 0.8868971 7.105350 0.09481398 0.001685054 0.03494056
## 41 10.769752 0.8710683 7.644010 0.17615521 0.003911593 0.07164127
## 42 10.732712 0.8719881 7.623433 0.11491391 0.002910961 0.08143229
## 43 10.718366 0.8723089 7.614080 0.11889379 0.002895929 0.07052008
## 44 10.714160 0.8725090 7.576604 0.10908784 0.001804921 0.06814471
## 45 10.734832 0.8719989 7.580524 0.07706717 0.001617523 0.03949596
```

```
model$bestTune
```

```
##    nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
## 14     100        11 0.1     0              0.8                1         1
```

```
write.csv(file="grid_search_results.csv", x=model$results)
```

```
model$finalModel
```

```
## ##### xgb.Booster
## raw: 4 Mb
## call:
##   xgboost::xgb.train(params = list(eta = param$eta, max_depth = param$max_depth,
##     gamma = param$gamma, colsample_bytree = param$colsample_bytree,
##     min_child_weight = param$min_child_weight, subsample = param$subsample),
##     data = x, nrounds = param$nrounds, objective = "reg:linear")
## params (as set within xgb.train):
##   eta = "0.1", max_depth = "11", gamma = "0", colsample_bytree = "0.8", min_child_weight = "1", subsa
## xgb.attributes:
##   niter
## callbacks:
##   cb.print.evaluation(period = print_every_n)
## # of features: 14
## niter: 100
## nfeatures : 14
## xNames : score_17 score_16 score_15 score_14 score_11 score_13 score_10 score_12 score_8 score_9 scor
## problemType : Regression
## tuneValue :
##     nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
## 14     100        11 0.1     0              0.8                1         1
## obsLevels : NA
## param :
##  list()
```

```
y_pred <- predict(model,as.matrix(test_X))
residuals = test_y - y_pred
RMSE = sqrt(mean(residuals^2))
cat("residual mean squared error", RMSE, "\n")
```

```
## residual mean squared error 7.797923
```

```
sse <- sum((test_y - y_pred) ** 2)
tse <- sum((test_y - mean(test_y)) ** 2)
r_squared<-1-(sse/tse)

cat("r-squared", r_squared, "\n")
```

```
## r-squared 0.9309522
```

Feature Importance with Grid Search Model

```
xgb1 <- xgb.train(eta = 0.1, max_depth = 11, gamma = 0, colsample_bytree = 0.9, min_child_weight = 1, su
```

```
## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").
```

```
## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").
```

```
## [1]  train-rmse:64.011742
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:23.975122
## [21] train-rmse:11.029884
## [31] train-rmse:7.507892
## [41] train-rmse:6.651770
## [51] train-rmse:6.300044
## [61] train-rmse:6.085419
## [71] train-rmse:5.932196
## [81] train-rmse:5.789577
## [91] train-rmse:5.610059
## [100]    train-rmse:5.498309
```

```
important_Df <- xgb.importance(model=xgb1)
```

```
write.csv(file="feature_importance_post_gs.csv", x = important_Df)
```

Continued Exploration Into Feature Importance (not in report, personal curiosity)

```
# best 77 for score_17, eta =  0.005, 1000, 15
xgb1 <- xgb.train(eta = 0.001, max_depth = 15, gamma = 0, colsample_bytree = 0.9, min_child_weight = 1,
```

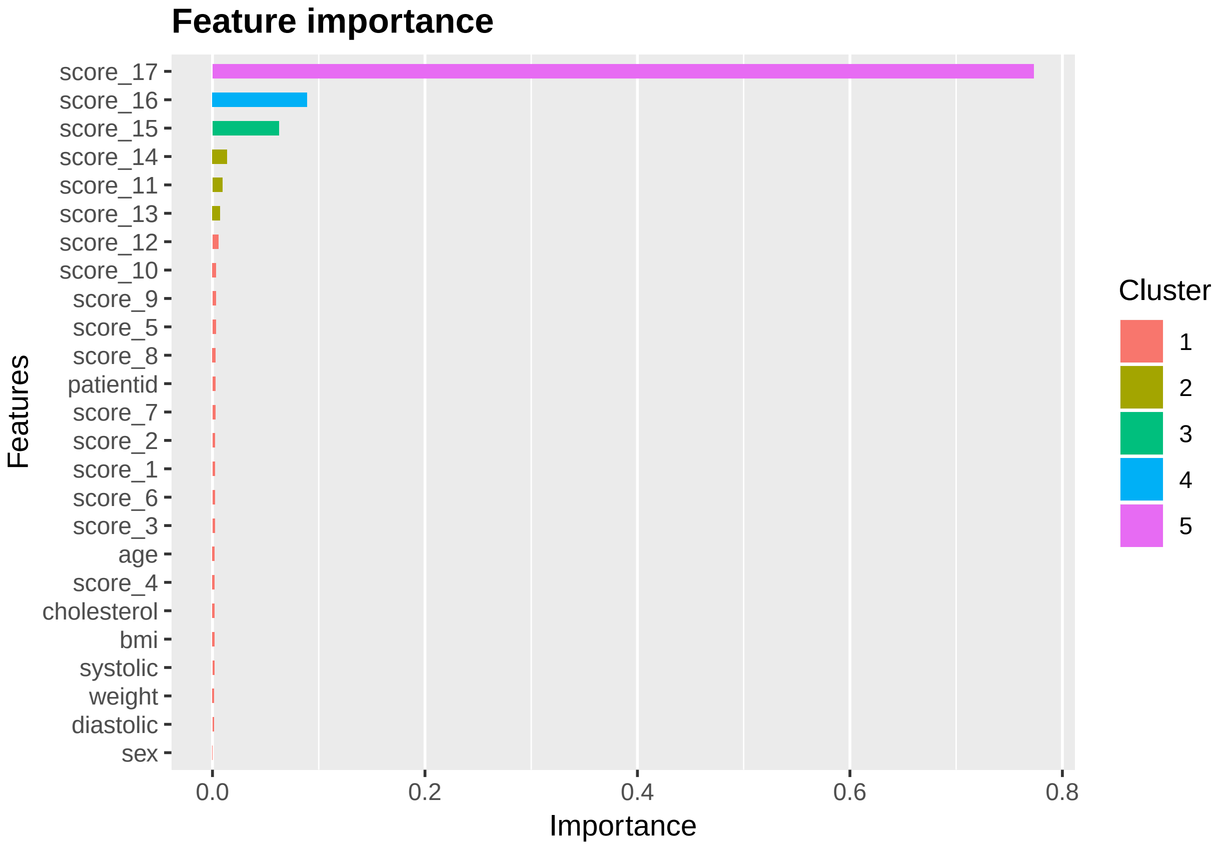
```
## Warning: 'print.every.n' is deprecated.
## Use 'print_every_n' instead.
## See help("Deprecated") and help("xgboost-deprecated").
```

```
## Warning: 'early.stop.round' is deprecated.
## Use 'early_stopping_rounds' instead.
## See help("Deprecated") and help("xgboost-deprecated").
```

```
## [1]  train-rmse:70.871452
## Will train until train_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:70.182121
## [21] train-rmse:69.499489
## [31] train-rmse:68.824074
## [41] train-rmse:68.154999
## [51] train-rmse:67.493301
## [61] train-rmse:66.839012
## [71] train-rmse:66.190918
## [81] train-rmse:65.548416
## [91] train-rmse:64.913101
## [101]    train-rmse:64.283997
## [111]    train-rmse:63.661480
## [121]    train-rmse:63.045845
## [131]    train-rmse:62.434917
## [141]    train-rmse:61.831120
## [151]    train-rmse:61.232899
## [161]    train-rmse:60.641659
## [171]    train-rmse:60.055550
## [181]    train-rmse:59.475269
## [191]    train-rmse:58.900673
## [201]    train-rmse:58.332020
## [211]    train-rmse:57.769501
## [221]    train-rmse:57.212582
## [231]    train-rmse:56.660862
## [241]    train-rmse:56.114998
## [251]    train-rmse:55.575523
## [261]    train-rmse:55.040306
## [271]    train-rmse:54.511505
## [281]    train-rmse:53.987560
## [291]    train-rmse:53.468624
## [301]    train-rmse:52.956535
## [311]    train-rmse:52.448498
## [321]    train-rmse:51.945412
## [331]    train-rmse:51.448158
## [341]    train-rmse:50.955200
## [351]    train-rmse:50.467617
## [361]    train-rmse:49.985306
## [371]    train-rmse:49.507381
## [381]    train-rmse:49.034866
## [391]    train-rmse:48.567722
## [401]    train-rmse:48.104572
## [411]    train-rmse:47.646561
## [421]    train-rmse:47.193676
## [431]    train-rmse:46.744446
## [441]    train-rmse:46.299809
## [451]    train-rmse:45.860424
## [461]    train-rmse:45.424625
## [471]    train-rmse:44.993134
## [481]    train-rmse:44.566227
## [491]    train-rmse:44.143456
## [501]    train-rmse:43.725819
## [511]    train-rmse:43.312759
## [521]    train-rmse:42.903534
## [531]    train-rmse:42.497826
## [541]    train-rmse:42.096764
## [551]    train-rmse:41.699989
## [561]    train-rmse:41.306927
## [571]    train-rmse:40.918148
## [581]    train-rmse:40.533863
## [591]    train-rmse:40.153049
## [601]    train-rmse:39.775764
## [611]    train-rmse:39.402328
## [621]    train-rmse:39.032902
## [631]    train-rmse:38.667217
## [641]    train-rmse:38.305450
## [651]    train-rmse:37.948353
## [661]    train-rmse:37.594460
## [671]    train-rmse:37.244026
## [681]    train-rmse:36.897614
## [691]    train-rmse:36.554569
## [701]    train-rmse:36.214478
## [711]    train-rmse:35.877712
## [721]    train-rmse:35.544582
## [731]    train-rmse:35.214699
## [741]    train-rmse:34.888371
## [751]    train-rmse:34.565445
## [761]    train-rmse:34.245899
## [771]    train-rmse:33.929298
## [781]    train-rmse:33.616322
## [791]    train-rmse:33.306355
## [801]    train-rmse:33.000134
## [811]    train-rmse:32.696453
## [821]    train-rmse:32.396187
## [831]    train-rmse:32.099018
## [841]    train-rmse:31.804802
## [851]    train-rmse:31.513487
## [861]    train-rmse:31.224821
## [871]    train-rmse:30.939514
## [881]    train-rmse:30.656996
## [891]    train-rmse:30.377878
## [901]    train-rmse:30.101542
## [911]    train-rmse:29.827442
## [921]    train-rmse:29.556494
## [931]    train-rmse:29.287954
## [941]    train-rmse:29.022163
## [951]    train-rmse:28.759302
## [961]    train-rmse:28.498922
## [971]    train-rmse:28.241661
## [981]    train-rmse:27.986969
## [991]    train-rmse:27.734583
## [1001]   train-rmse:27.485527
## [1011]   train-rmse:27.238173
## [1021]   train-rmse:26.993214
## [1031]   train-rmse:26.751448
## [1041]   train-rmse:26.512018
## [1051]   train-rmse:26.274883
## [1061]   train-rmse:26.040138
## [1071]   train-rmse:25.807384
## [1081]   train-rmse:25.577452
## [1091]   train-rmse:25.350113
## [1101]   train-rmse:25.124340
## [1111]   train-rmse:24.900785
## [1121]   train-rmse:24.679831
## [1131]   train-rmse:24.460876
## [1141]   train-rmse:24.244127
## [1151]   train-rmse:24.029995
## [1161]    train-rmse:23.817511
## [1171]    train-rmse:23.607548
## [1181]    train-rmse:23.399286
## [1191]    train-rmse:23.194075
## [1201]    train-rmse:22.990013
## [1211]    train-rmse:22.787870
## [1221]    train-rmse:22.588333
## [1231]    train-rmse:22.390463
## [1241]    train-rmse:22.194609
## [1251]    train-rmse:22.000666
## [1261]    train-rmse:21.808775
## [1271]    train-rmse:21.618744
## [1281]    train-rmse:21.430931
## [1291]    train-rmse:21.245205
## [1301]    train-rmse:21.060812
## [1311]    train-rmse:20.878099
## [1321]    train-rmse:20.697470
## [1331]    train-rmse:20.518930
## [1341]    train-rmse:20.341715
## [1351]    train-rmse:20.166164
## [1361]    train-rmse:19.992418
## [1371]    train-rmse:19.820293
## [1381]    train-rmse:19.650291
## [1391]    train-rmse:19.481920
## [1401]    train-rmse:19.314964
## [1411]    train-rmse:19.149776
## [1421]    train-rmse:18.986603
## [1431]    train-rmse:18.824808
## [1441]    train-rmse:18.664303
## [1451]    train-rmse:18.505653
## [1461]    train-rmse:18.348436
## [1471]    train-rmse:18.192535
## [1481]    train-rmse:18.038691
## [1491]    train-rmse:17.885851
## [1501]    train-rmse:17.734653
## [1511]    train-rmse:17.584997
## [1521]    train-rmse:17.437199
## [1531]    train-rmse:17.290985
## [1541]    train-rmse:17.146044
## [1551]    train-rmse:17.002434
## [1561]    train-rmse:16.860670
## [1571]    train-rmse:16.719566
## [1581]    train-rmse:16.579962
## [1591]    train-rmse:16.441782
## [1601]    train-rmse:16.305151
## [1611]    train-rmse:16.169376
## [1621]    train-rmse:16.035156
## [1631]    train-rmse:15.902055
## [1641]    train-rmse:15.770550
## [1651]    train-rmse:15.640607
## [1661]    train-rmse:15.511923
## [1671]    train-rmse:15.384394
## [1681]    train-rmse:15.258030
## [1691]    train-rmse:15.133065
## [1701]	train-rmse:15.009680
## [1711]	train-rmse:14.886911
## [1721]	train-rmse:14.765257
## [1731]	train-rmse:14.645016
## [1741]	train-rmse:14.525983
## [1751]	train-rmse:14.407823
## [1761]	train-rmse:14.291235
## [1771]	train-rmse:14.176161
## [1781]	train-rmse:14.062056
## [1791]	train-rmse:13.948627
## [1801]	train-rmse:13.836627
## [1811]	train-rmse:13.725524
## [1821]	train-rmse:13.615478
## [1831]	train-rmse:13.506728
## [1841]	train-rmse:13.398695
## [1851]	train-rmse:13.291558
## [1861]	train-rmse:13.185975
## [1871]	train-rmse:13.081483
## [1881]	train-rmse:12.978724
## [1891]	train-rmse:12.876172
## [1901]	train-rmse:12.774461
## [1911]	train-rmse:12.673843
## [1921]	train-rmse:12.574474
## [1931]	train-rmse:12.475921
## [1941]	train-rmse:12.377992
## [1951]	train-rmse:12.281311
## [1961]	train-rmse:12.185540
## [1971]	train-rmse:12.090712
## [1981]	train-rmse:11.997290
## [1991]	train-rmse:11.904860
## [2001]	train-rmse:11.812653
## [2011]	train-rmse:11.721831
## [2021]	train-rmse:11.632300
## [2031]	train-rmse:11.543447
## [2041]	train-rmse:11.455143
## [2051]	train-rmse:11.367746
## [2061]	train-rmse:11.280928
## [2071]	train-rmse:11.195229
## [2081]	train-rmse:11.110668
## [2091]	train-rmse:11.026523
## [2101]	train-rmse:10.943012
## [2111]	train-rmse:10.860858
## [2121]	train-rmse:10.779476
## [2131]	train-rmse:10.698826
## [2141]	train-rmse:10.619030
## [2151]	train-rmse:10.540205
## [2161]	train-rmse:10.461802
## [2171]	train-rmse:10.384429
## [2181]	train-rmse:10.307970
## [2191]	train-rmse:10.231699
## [2201]	train-rmse:10.156730
## [2211]	train-rmse:10.083035
## [2221]	train-rmse:10.009233
## [2231]	train-rmse:9.936252
## [2241]   train-rmse:9.864209
## [2251]   train-rmse:9.793258
## [2261]   train-rmse:9.722599
## [2271]   train-rmse:9.652504
## [2281]   train-rmse:9.583209
## [2291]   train-rmse:9.515148
## [2301]   train-rmse:9.447438
## [2311]   train-rmse:9.380944
## [2321]   train-rmse:9.314778
## [2331]   train-rmse:9.248869
## [2341]   train-rmse:9.184072
## [2351]   train-rmse:9.119517
## [2361]   train-rmse:9.056173
## [2371]   train-rmse:8.993618
## [2381]   train-rmse:8.931767
## [2391]   train-rmse:8.870363
## [2401]   train-rmse:8.809480
## [2411]   train-rmse:8.749520
## [2421]   train-rmse:8.689697
## [2431]   train-rmse:8.630487
## [2441]   train-rmse:8.572026
## [2451]   train-rmse:8.513571
## [2461]   train-rmse:8.456100
## [2471]   train-rmse:8.399267
## [2481]   train-rmse:8.343366
## [2491]   train-rmse:8.287911
## [2501]   train-rmse:8.232965
## [2511]   train-rmse:8.178582
## [2521]   train-rmse:8.125272
## [2531]   train-rmse:8.072109
## [2541]   train-rmse:8.019469
## [2551]   train-rmse:7.967668
## [2561]   train-rmse:7.916230
## [2571]   train-rmse:7.864948
## [2581]   train-rmse:7.814294
## [2591]   train-rmse:7.764372
## [2601]   train-rmse:7.714857
## [2611]   train-rmse:7.665568
## [2621]   train-rmse:7.617573
## [2631]   train-rmse:7.570236
## [2641]   train-rmse:7.523200
## [2651]   train-rmse:7.477261
## [2661]   train-rmse:7.431259
## [2671]   train-rmse:7.386332
## [2681]   train-rmse:7.340892
## [2691]   train-rmse:7.296271
## [2701]   train-rmse:7.252033
## [2711]   train-rmse:7.208269
## [2721]   train-rmse:7.165020
## [2731]   train-rmse:7.121891
## [2741]   train-rmse:7.079193
## [2751]   train-rmse:7.037067
## [2761]   train-rmse:6.995992
## [2771]   train-rmse:6.954371
## [2781]   train-rmse:6.913537
## [2791]   train-rmse:6.873390
## [2801]   train-rmse:6.833553
## [2811]   train-rmse:6.794659
## [2821]   train-rmse:6.756043
## [2831]   train-rmse:6.717995
## [2841]   train-rmse:6.679582
## [2851]   train-rmse:6.641189
## [2861]   train-rmse:6.603414
## [2871]   train-rmse:6.565611
## [2881]   train-rmse:6.528114
## [2891]   train-rmse:6.491240
## [2901]   train-rmse:6.455054
## [2911]   train-rmse:6.419475
## [2921]   train-rmse:6.383976
## [2931]   train-rmse:6.348577
## [2941]   train-rmse:6.313704
## [2951]   train-rmse:6.279154
## [2961]   train-rmse:6.245039
## [2971]   train-rmse:6.210994
## [2981]   train-rmse:6.177809
## [2991]   train-rmse:6.145113
## [3000]   train-rmse:6.115584
```

```
important_Df <- xgb.importance(model=xgb1)
```

```
xgb.ggplot.importance(importance_matrix=important_Df)
```



```
important_Df[0:3,c("Feature","Gain")]
```

```
##     Feature       Gain
## 1: score_17 0.77289117
## 2: score_16 0.08910719
## 3: score_15 0.06238864
```

```
best_score<-xgb1$best_score

cat("best residual mean squared error", best_score, "\n")
```

```
## best residual mean squared error 6.115584
```

```
y_pred <- predict(xgb1,as.matrix(val_X))
sse <- sum((val_y - y_pred) ** 2)
tse <- sum((val_y - mean(val_y)) ** 2)
r_squared<-1-(sse/tse)

cat("r-squared", r_squared, "\n")
```

```
## r-squared 0.8815765
```

0.84173559