> inTrain <- createDataPartition(data$target, p=.8)[[1]]

> dataTr <- data[inTrain,]

> dataTs <- data[-inTrain,]

> library(lightgbm)

> # lgb dataset

> train\_s <- as.matrix(dataTr[,-4])

> dtrain <- lgb.Dataset(data=train\_s, label = dataTr$target)

>

> test\_s <- as.matrix(dataTs[,-4])

> library(lightgbm)

> # lgb dataset

> train\_s <- as.matrix(dataTr[,-4])

> dtrain <- lgb.Dataset(data=train\_s, label = dataTr$target)

>

> test\_s <- as.matrix(dataTs[,-4])

>

>

> params <- list(objective="regression",

+ metric = "l2",

+ # min\_sum\_hessian\_in\_leaf = 1,

+ feature\_fraction = 0.7,

+ bagging\_fraction = 0.7,

+ bagging\_freq = 5,

+ # min\_data = 100,

+ max\_bin = 50,

+ lambda\_l1 = 8,

+ lambda\_l2 = 1.3

+ # min\_data\_in\_bin=100,

+ # min\_gain\_to\_split = 10,

+ # min\_data\_in\_leaf = 30

+ #is\_unbalance = TRUE

+ )

>

> # lgb.model.cv <- lgb.cv(params = lgb.grid,

> # data = dtrain,

> # learning\_rate = 0.02,

> # #num\_leaves = 25,

> # num\_threads = 2 ,

> # nrounds = 500,

> # early\_stopping\_rounds = 30,

> # eval\_freq = 20,

> # #eval = lgb.normalizedgini,

> # #categorical\_feature = categoricals.vec,

> # nfold = 5,

> # stratified = TRUE)

>

> # best.iter = lgb.model.cv$best\_iter

> lgb\_m <- lgb.train(params=params,

+ data=dtrain,

+ min\_data =1,

+ learning\_rate=0.1,

+ nrounds = 500)

[LightGBM] [Info] Total Bins 3780

[LightGBM] [Info] Number of data: 161536, number of used features: 112

[LightGBM] [Info] Start training from score -0.401036

> preds\_lgb <- predict(lgb\_m, test\_s)

> summary(preds\_lgb)

Min. 1st Qu. Median Mean 3rd Qu. Max.

-12.1063 -0.6078 -0.3475 -0.4056 -0.1152 1.3206

> rmse(dataTs$target,preds\_lgb)

[1] 3.718194