f. Fine tune the model and represent important features

# create design matrices

train\_x = model.Matrix(V281 ~ . - 1, data = train, sparse = F)

train\_y = train$V281

test\_x = model.Matrix(V281 ~ . - 1, data = test, sparse = F)

test\_y = test$V281

train\_xgb = xgb.DMatrix(data = as.matrix(train\_x), label = train\_y)

test\_xgb = xgb.DMatrix(data = as.matrix(test\_x), label = test\_y)

g. Interpret the summary of the linear model

# fit XGBoost

pred\_xgb = foreach(i = 1:n, .combine = cbind) %do% {

mdl\_xgb = xgboost(data = train\_xgb, nround = 750, nthread = 4, max\_depth = 6, eta = 0.025, subsample = 0.7, gamma = 3)

return(predict(mdl\_xgb, test\_xgb))

}

h. Report the test accuracy vs. the training accuracy

# fit random forest

pred\_rf = foreach(i = 1:n, .combine = cbind) %do% {

mdl\_rf = ranger(V281 ~ ., data = train, num.trees = 1000, mtry = 120, write.forest = T)

return(predict(mdl\_rf, test)$predictions)

}

i. Interpret the final model coefficients

j. Plot the model result and compare it with assumptions of the model