Model Results

Nick Wawee

4/6/2021

## Overall Results Plot

### Input Data

rn = c('Accuracy', 'Sensitivity', 'Specificity', 'Kappa')  
  
#all samples  
df\_all = data.frame('Stepwise Logit'= c(0.8644,0.7923, 0.8780, 0.5689),  
 'CCT' = c(0.8701, 0.8249,0.8786, 0.5911),  
 'Neural Network' = c(0.9077, 0.8872, 0.9116,0.698),  
 'KNN' = c(0.8781, 0.3947,0.9692, 0.4434),  
 'Random Forest' = c(0.956, 0.819, 0.982, 0.83),  
 'ADABoost'= c(0.968, 0.875, 0.986, 0.879))  
  
row.names(df\_all) = rn  
df\_all$Metric = rn  
df\_all.m = melt(df\_all)

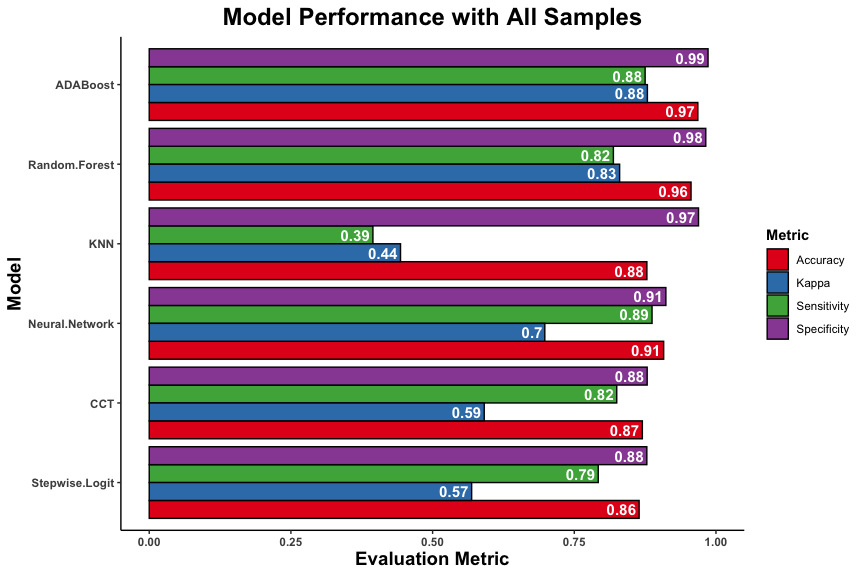
## Using Metric as id variables

#undersample  
  
df\_under = data.frame('Stepwise Logit'= c(0.8013, 0.8813, 0.7862, 0.4729),  
 'CCT' = c(0.8484, 0.9080,0.8372, 0.5675),  
 'Neural Network' = c(0.8202, 0.8961, 0.8058,0.5103),  
 'KNN' = c(0.8084, 0.7923, 0.8114, 0.4569),  
 'Random Forest' = c(0.924, 0.944, 0.921, 0.753),  
 'ADABoost'= c(0.938, 0.953, 0.935, 0.792))  
row.names(df\_under) = rn  
df\_under$Metric = rn  
df\_under.m = melt(df\_under)

## Using Metric as id variables

### All Sample Plot

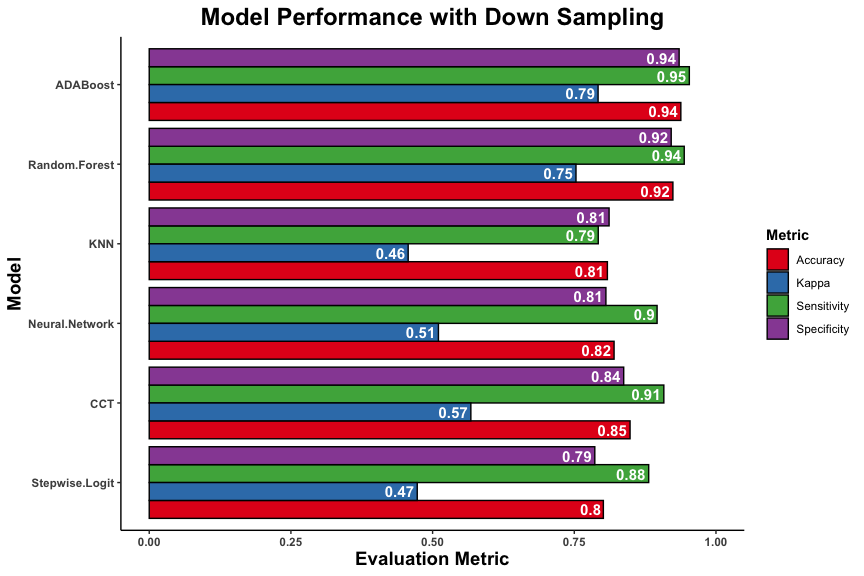
df\_all.m$Labels = round(df\_all.m$value, 2)  
p = ggplot(data = df\_all.m, aes(y = value, x = variable, fill = Metric))+  
 geom\_col(position=position\_dodge(), colour="black")+labs(x = 'Model', y='Evaluation Metric', title = 'Model Performance with All Samples')+ geom\_text(aes(label=Labels), position = position\_dodge(width=0.9),color="white", size=4, hjust = 1.1, fontface = 'bold')+plot\_opts+theme(axis.text.x = element\_text(face = 'bold'), axis.text.y = element\_text(face = 'bold'), legend.title = element\_text(face = 'bold'))+coord\_flip()+scale\_fill\_brewer(palette = 'Set1')+ylim(0,1)  
p



ggsave('../plots/allsample\_metrics.png',plot = p, device = 'png', height = 6, width = 9, units = 'in', dpi = 600)

### Undersample Plot

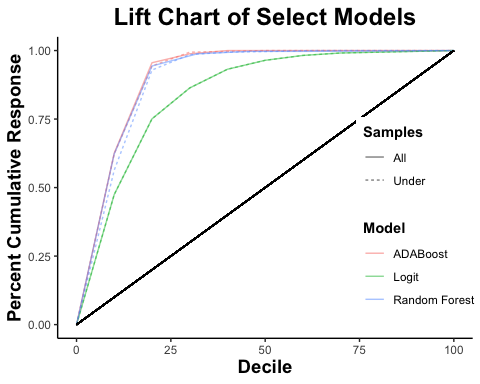
df\_under.m$Labels = round(df\_under.m$value, 2)  
p = ggplot(data = df\_under.m, aes(y = value, x = variable, fill = Metric))+  
 geom\_col(position=position\_dodge(), colour="black")+labs(x = 'Model', y='Evaluation Metric', title = 'Model Performance with Down Sampling')+ geom\_text(aes(label=Labels), position = position\_dodge(width=0.9),color="white", size=4, hjust = 1.1, fontface = 'bold')+plot\_opts+theme(axis.text.x = element\_text(face = 'bold'), axis.text.y = element\_text(face = 'bold'), legend.title = element\_text(face = 'bold'))+coord\_flip()+scale\_fill\_brewer(palette = 'Set1')+ylim(0,1)  
p



ggsave('../plots/undersample\_metrics.png',plot = p, device = 'png', height = 6, width = 9, units = 'in', dpi = 600)

### Lift Charts

library(gains)  
  
##loading probabilities  
  
test.data = read.csv('../data/processed/test.csv')  
test.data$Attrition\_Flag = ifelse(test.data$Attrition\_Flag == "Attrited Customer",1 ,0)  
  
pred.prob.ada = read.csv('../data/processed/ada\_prob.csv')  
pred.prob.ada.under = read.csv('../data/processed/ada\_prob\_under.csv')  
  
pred.prob.rf = read.csv('../data/processed/rf\_prob.csv')  
pred.prob.rf.under = read.csv('../data/processed/rf\_prob\_under.csv')  
  
pred.prob.logit = read.csv('../data/processed/logit\_prob.csv')  
pred.prob.logit.under = read.csv('../data/processed/logit\_prob\_under.csv')  
  
##ada  
gain <- gains(test.data$Attrition\_Flag, pred.prob.ada$Attrited.Customer)   
x.ada <- c(0, gain$depth)  
pred.y.ada <- c(0, gain$cume.pct.of.total)  
adadf = data.frame('Percent\_Cumulative\_Response' = pred.y.ada, 'Decile' = x.ada)  
adadf$Model = 'ADABoost'  
adadf$Samples = 'All'  
  
gain <- gains(test.data$Attrition\_Flag, pred.prob.ada.under$Attrited.Customer)   
x.ada <- c(0, gain$depth)  
pred.y.ada <- c(0, gain$cume.pct.of.total)  
adadf\_under = data.frame('Percent\_Cumulative\_Response' = pred.y.ada, 'Decile' = x.ada)  
adadf\_under$Model = 'ADABoost'  
adadf\_under$Samples = 'Under'  
  
##random forest  
gain <- gains(test.data$Attrition\_Flag, pred.prob.rf$Attrited.Customer)   
x.rf <- c(0, gain$depth)  
pred.y.rf <- c(0, gain$cume.pct.of.total)  
rfdf = data.frame('Percent\_Cumulative\_Response' = pred.y.rf, 'Decile' = x.rf)  
rfdf$Model = 'Random Forest'  
rfdf$Samples = 'All'  
  
gain <- gains(test.data$Attrition\_Flag, pred.prob.rf.under$Attrited.Customer)   
x.rf <- c(0, gain$depth)  
pred.y.rf <- c(0, gain$cume.pct.of.total)  
rfdf\_under = data.frame('Percent\_Cumulative\_Response' = pred.y.rf, 'Decile' = x.rf)  
rfdf\_under$Model = 'Random Forest'  
rfdf\_under$Samples = 'Under'  
  
##logit  
gain <- gains(test.data$Attrition\_Flag, pred.prob.logit$Attrited.Customer)   
x.logit <- c(0, gain$depth)  
pred.y.logit <- c(0, gain$cume.pct.of.total)  
logitdf = data.frame('Percent\_Cumulative\_Response' = pred.y.logit, 'Decile' = x.logit)  
logitdf$Model = 'Logit'  
logitdf$Samples = 'All'  
  
gain <- gains(test.data$Attrition\_Flag, pred.prob.logit.under$Attrited.Customer)   
x.ada <- c(0, gain$depth)  
pred.y.ada <- c(0, gain$cume.pct.of.total)  
logitdf\_under = data.frame('Percent\_Cumulative\_Response' = pred.y.logit, 'Decile' = x.logit)  
logitdf\_under$Model = 'Logit'  
logitdf\_under$Samples = 'Under'  
  
#combining all dfs  
liftchartdf = rbind(adadf, adadf\_under,rfdf, rfdf\_under, logitdf, logitdf\_under)  
  
liftp = ggplot(data = liftchartdf, aes(x = Decile, y = Percent\_Cumulative\_Response))+  
 geom\_line(aes(col= Model, linetype = Samples), alpha = 0.5)+  
 plot\_opts+labs(title = 'Lift Chart of Select Models', y = 'Percent Cumulative Response')+  
 theme(legend.title = element\_text(face = 'bold'),legend.position = c(0.87, 0.4))+  
 geom\_segment(aes(x = 0, xend = 100, y = 0, yend =1))  
liftp

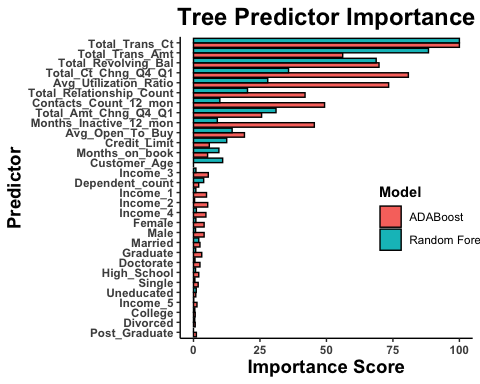


ggsave('../plots/liftchart.png',plot = liftp, device = 'png', height = 5, width = 6, units = 'in', dpi = 600)

### Variable Importance

#### Trees

load('../data/processed/vimps\_trees.rda')  
vimpdf$Labels = round(vimpdf$Overall,2)  
  
#finding mean value for order  
meandf = data.frame('Mean' = as.numeric())  
for (l in levels(as.factor(vimpdf$Predictor))){  
 meandf[l, 'Mean'] = mean(vimpdf$Overall[vimpdf$Predictor == l])  
}  
  
meandf = as.matrix(meandf)  
meandf = meandf[order(meandf, decreasing = F),]  
meandf = data.frame(imp = meandf)  
meandf$o = seq(from = 1, to = nrow(meandf), by = 1)  
vimpdf$o[match(rownames(meandf), vimpdf$Predictors)] = meandf$o  
  
treeimp\_p = vimpdf %>%  
 mutate(Predictors = fct\_reorder(Predictors, o)) %>%  
 ggplot( aes(y = Overall, x = Predictors, fill = Model))+  
 geom\_col(position = position\_dodge(), colour="black")+labs(x = 'Predictor', y='Importance Score', title = 'Tree Predictor Importance')+ plot\_opts+theme(axis.text.x = element\_text(face = 'bold'), axis.text.y = element\_text(face = 'bold'), legend.title = element\_text(face = 'bold'), legend.position = c(0.87, 0.4))+coord\_flip()  
  
treeimp\_p



ggsave('../plots/treepredimp\_under.png',plot = treeimp\_p, device = 'png', height = 9, width = 6, units = 'in', dpi = 600)

#### Stepwise Logit

load('../data/processed/logit\_under.rda')  
  
logit = step\_cv$finalModel  
  
png('../plots/logit\_under\_coeffs.png', height = 4, width = 6, res = 600, units = 'in')  
plot\_coefs(logit, exp= T)

## Registered S3 methods overwritten by 'broom':  
## method from   
## tidy.glht jtools  
## tidy.summary.glht jtools

## Loading required namespace: broom.mixed

## Registered S3 method overwritten by 'broom.mixed':  
## method from   
## tidy.gamlss broom

dev.off()

## quartz\_off\_screen   
## 2

tbl = tidy(logit)  
tbl$estimate = round(tbl$estimate,4)  
tbl$std.error = round(tbl$std.error,4)  
tbl$statistic = round(tbl$statistic,4)  
tbl$p.value = round(tbl$p.value,4)  
  
write.table(tbl, file = '../data/processed/logit\_summary.txt', sep = ",", quote = FALSE, row.names = F)