```
#########################################################################
########
#
# Replication Data Political Analysis Forum: Comparing Random Forest with
# Logistic Regression for Predicting Class-Imbalanced Civil War Onset Data
#
#########################################################################
########

#Set the Working Directory
setwd("Desktop/RF_Replication_2018_PA_Forum")
getwd()
# data for prediction
data=read.csv(file="SambanisImp.csv")
# data for looking at Variable Importance Plots
data2<-read.csv(file="Amelia.Imp3.csv")

library(randomForest) #for random forests
library(caret) # for CV folds and data splitting
library(ROCR) # for diagnostics and ROC plots/stats
library(pROC) # same as ROCR
library(stepPlr) # Firth's logit implemented thru caret library
library(doMC) # for using multiple processor cores
library(xtable) # for writing Table 1 in Latex

###Use only the 88 variables specified in Sambanis (2006) Appendix###
data.full<-data[,c("warstds", "ager", "agexp", "anoc", "army85", "autch98", "auto4",
"autonomy", "avgnabo", "centpol3", "coldwar", "decade1", "decade2",
"decade3", "decade4", "dem", "dem4", "demch98", "dlang", "drel",
"durable", "ef", "ef2", "ehet", "elfo", "elfo2", "etdo4590",
"expgdp", "exrec", "fedpol3", "fuelexp", "gdpgrowth", "geo1", "geo2",
"geo34", "geo57", "geo69", "geo8", "illiteracy", "incumb", "infant",
"inst", "inst3", "life", "lmtnest", "ln_gdpen", "lpopns", "major", "manuexp", "milper",
"mirps0", "mirps1", "mirps2", "mirps3", "nat_war", "ncontig",
"nmgdp", "nmdp4_alt", "numlang", "nwstate", "oil", "p4mchg",
"parcomp", "parreg", "part", "partfree", "plural", "plurrel",
"pol4", "pol4m", "pol4sq", "polch98", "polcomp", "popdense",
"presi", "pri", "proxregc", "ptime", "reg", "regd4_alt", "relfrac", "seceduc",
"second", "semipol3", "sip2", "sxpnew", "sxpsq", "tnatwar", "trade",
"warhist", "xconst")]

###Convert DV into Factor with names for Caret Library###
data.full$warstds<-factor(
data.full$warstds,
levels=c(0,1),
labels=c("peace", "war"))
```

```
# distribute workload over multiple cores for faster computation
registerDoMC(cores=7)

set.seed(666)



# This is the cross-validation function for the Caret Library.
# The code savePredictions=T has been changed from the 2015 code. This allows the
user to
# extract predicted probabilities directly from the best cross-validated model.
# This was Wang's critique of Figures 1 and 2, which were overfit.
# Though the AUC scores reported were accurate, the ROC curves did not match the
# reported AUC values. This has now been corrected so that the revised figures will
have the
# correctly matching curves to the AUC values.

tc<-trainControl(method="cv",
number=10,
summaryFunction=twoClassSummary,
classProb=T,
savePredictions = T)


#Fearon and Laitin Model (2003) Specification###
model.fl.1<-
train(as.factor(warstds)~warhist+ln_gdpen+lpopns+lmtnest+ncontig+oil+nwstate
+inst3+pol4+ef+relfrac, #FL 2003 model spec
metric="ROC", method="glm", family="binomial",
trControl=tc, data=data.full)

summary(model.fl.1)
model.fl.1


### Fearon and Laitin (2003) penalized logistic regression
model.fl.2<-
train(as.factor(warstds)~warhist+ln_gdpen+lpopns+lmtnest+ncontig+oil+nwstate
+inst3+pol4+ef+relfrac, #FL 2003 model spec
metric="ROC", method="plr",
trControl=tc, data=data.full)
summary(model.fl.2)
model.fl.2


### Collier and Hoeffler (2004) Model specification###
```

```
model.ch.1<-
train(as.factor(warstds)~sxpnew+sxpsq+ln_gdpen+gdpgrowth+warhist+lmtnest+ef+pop
dense
+lpopns+coldwar+seceduc+ptime,
metric="ROC", method="glm", family="binomial",
trControl=tc, data=data.full)
model.ch.1


### Collier and Hoeffler penalized logistic regression###

model.ch.2<-
train(as.factor(warstds)~sxpnew+sxpsq+ln_gdpen+gdpgrowth+warhist+lmtnest+ef+pop
dense
+lpopns+coldwar+seceduc+ptime,
metric="ROC", method="plr",
trControl=tc, data=data.full)
model.ch.2


### Hegre and Sambanis (2006) Model Specification###

model.hs.1<-
train(warstds~lpopns+ln_gdpen+inst3+parreg+geo34+proxregc+gdpgrowth+anoc+
partfree+nat_war+lmtnest+decade1+pol4sq+nwstate+regd4_alt+etdo4590+milper+
geo1+tnatwar+presi,
metric="ROC", method="glm", family="binomial",
trControl=tc, data=data.full)
model.hs.1


model.hs.2<-
train(warstds~lpopns+ln_gdpen+inst3+parreg+geo34+proxregc+gdpgrowth+anoc+
partfree+nat_war+lmtnest+decade1+pol4sq+nwstate+regd4_alt+etdo4590+milper+
geo1+tnatwar+presi,
metric="ROC", method="plr",
trControl=tc, data=data.full)
model.hs.2


### Random Forest ###

model.rf<-train(as.factor(warstds)~.,
metric="ROC", method="rf",
sampsize=c(30,90),
importance=T,
```

```
proximity=F, ntree=1000,
trControl=tc, data=data.full)
model.rf


#########################################################################
############################

### Random Forest without CV to get OOB Error Rate - see footnote 7###

# RF.out<-randomForest(as.factor(warstds)~., sampsize=c(30, 90),
# importance=T, proximity=F, ntree=1000, confusion=T, err.rate=T, data=data.full)
# print(RF.out)


###Random Forests on Amelia Imputed Data for Variable Importance Plot###
###Data Imputed only for Theoretically Important Variables### Done to analyze variable
importance
###Data already read in on line 14 at the start of this file:
###data2<-read.csv(file="Amelia.Imp3.csv") ###

# myvars <- names(data2) %in% c("X", "country", "year", "atwards")
# newdata <- data2[!myvars]

# RF.out.am<-randomForest(as.factor(warstds)~.,sampsize=c(30, 90),
# importance=T, proximity=F, ntree=1000, confusion=T, err.rate=T, data=newdata)

# varImpPlot(RF.out.am, sort=T, type=2, main="Variables Contributing Most to
Predictive Accuracy of
# Random Forests",n.var=20)


###Creating dotplot for Variable Importance Plot for RF, results shown on page 13 ###
# importance(RF.out.am)

# x<-c(2.3246380, 2.3055470, 1.8544127, 1.7447636, 1.6848230, 1.6094923,
# 1.5564487, 1.4832437, 1.4100489, 1.3116247, 1.2875924, 1.1799487,
# 1.1034743, 1.0983414, 1.0689367, 1.0663479, 1.0123892, 0.9961138, 0.9961138,
0.9922545)
# g<-c("GDP Growth", "GDP per Capita",
# "Life Expectancy", "Western Europe and US Dummy", "Infant Mortality",
# "Trade as Percent of GDP", "Mountainous Terrain", "Illiteracy Rate",
# "Population (logged)", "Linguistic Heterogeneity", "Anocracy", "Median Regional Polity
Score",
# "Primary Commodity Exports (Squared)", "Democracy", "Military Power", "Population
Density",
# "Political Instability", "Ethnic Fractionalization", "Secondary Education",
```

```
# "Primary Commodity Exports")
# dotchart(rev(x), rev(g), cex=0.75, xlab="Mean Decrease in Gini Score (OOB
Estimates)",
# main="Variable Importance for Random Forests",
# xlim=c(1, 2.5))
#########################################################################
################################
```

###ROC Plots for Different Models###

```
library(ROCR)
attach(data.full)

#### This is the corrected code that generates the predicted probabilities from the
model
### with the highest AUC score in the caret CV procedure.

pred.FL.war<-model.fl.1$finalModel$fitted.values
pred.CH.war<-model.ch.1$finalModel$fitted.values
pred.HR.war<-model.hs.1$finalModel$fitted.values

### Notice the key difference between original code and revised code is the
### $finalModel$fitted.values this extracts the predicted probabilities
### from the best caret CV model, ensuring ROC curves drawn will match AUC scores.


### predicted probabilities for the Random Forest model
RF.1.pred<-predict(model.rf$finalModel, type="prob")
RF.1.pred<-as.data.frame(RF.1.pred)



### Plot ROC Curves (Corrected)### originally shown on Figure 2 page 10
pred.FL <- prediction(pred.FL.war, data.full$warstds)
perf.FL <- performance(pred.FL,"tpr","fpr")
pred.CH <- prediction(pred.CH.war, data.full$warstds)
perf.CH <- performance(pred.CH,"tpr","fpr")
pred.HS<-prediction(pred.HR.war, data.full$warstds)
perf.HS<-performance(pred.HS, "tpr", "fpr")
pred.RF.1<-prediction(RF.1.pred$war, data.full$warstds)
perf.RF.1<-performance(pred.RF.1, "tpr", "fpr")


### Code for plotting the corrected ROC Curves in Figure 1.

plot(perf.FL, main="Uncorrected Logits and Random Forests (Corrected)")
```

```
plot(perf.CH, add=T, lty=2)
plot(perf.HS, add=T, lty=3)
plot(perf.RF.1, add=T, lty=4)
legend(0.32, 0.25, c("Fearon and Laitin (2003) 0.77", "Collier and Hoeffler (2004) 0.82",
"Hegre and Sambanis (2006) 0.80", "Random Forest 0.91" ), lty=c(1,2,3,4), bty="n",
cex = .75)
```

The code to correct the Penalized Logistic regression figure in Figure 2 is shown below.

```
### ROC Plots for Penalized Logits and RF###
FL.2.pred<-1-model.fl.2$finalModel$fitted.values
CH.2.pred<-1-model.ch.2$finalModel$fitted.values
HS.2.pred<-1-model.hs.2$finalModel$fitted.values


pred.FL.2 <- prediction(FL.2.pred, data.full$warstds)
perf.FL.2 <- performance(pred.FL.2,"tpr","fpr")
pred.CH.2<- prediction(CH.2.pred, data.full$warstds)
perf.CH.2 <- performance(pred.CH.2,"tpr","fpr")
pred.HS.2<- prediction(HS.2.pred, data.full$warstds)
perf.HS.2 <- performance(pred.HS.2,"tpr","fpr")



#### Plot corrected ROC Curves in Figure 1 for penalized logistic regression models.
plot(perf.FL.2, main="Penalized Logits and Random Forests (Corrected)")
plot(perf.CH.2, add=T, lty=2)
plot(perf.HS.2, add=T, lty=3)
plot(perf.RF.1, add=T, lty=4)
legend(0.32, 0.25, c("Fearon and Laitin (2003) 0.77", "Collier and Hoeffler (2004) 0.77",
"Hegre and Sambanis (2006) 0.80", "Random Forest 0.91" ), lty=c(1,2,3,4), bty="n",
cex = .75)

### Combine both ROC plots

par(mfrow=c(1,2))

plot(perf.FL, main="Logits and Random Forests (Corrected)")
plot(perf.CH, add=T, lty=2)
plot(perf.HS, add=T, lty=3)
plot(perf.RF.1, add=T, lty=4)
legend(0.32, 0.25, c("Fearon and Laitin (2003) 0.77", "Collier and Hoeffler (2004) 0.82",
"Hegre and Sambanis (2006) 0.80", "Random Forest 0.91" ), lty=c(1,2,3,4), bty="n",
cex = .75)
```

```
plot(perf.FL.2, main="Penalized Logits and Random Forests (Corrected)")
plot(perf.CH.2, add=T, lty=2)
plot(perf.HS.2, add=T, lty=3)
plot(perf.RF.1, add=T, lty=4)
legend(0.32, 0.25, c("Fearon and Laitin (2003) 0.77", "Collier and Hoeffler (2004) 0.77",
"Hegre and Sambanis (2006) 0.80", "Random Forest 0.91" ), lty=c(1,2,3,4), bty="n",
cex = .75)



### Corrected code to draw corrected Separation Plots in Figure 2 as per Wang.

###Separation Plots###
library(separationplot)

## Transform DV back to 0,1 values for separation plots.
data.full$warstds<-factor(
data.full$warstds,
levels=c("peace","war"),
labels=c(0, 1))

#transform actual observations into vector for separation plots.
Warstds<-as.vector(data.full$warstds)



###Corrected Separation Plots###

### The corrections are the extraction of the fitted values for the logistic regression
models
### from caret CV procedure.

separationplot(RF.1.pred$war, Warstds, type = "line", line = T, lwd2=1,
show.expected=T,
heading="Random Forests", height=2.5, col0="white", col1="black")

separationplot(pred.FL.war, Warstds, type = "line", line = T, lwd2=1, show.expected=T,
heading="Fearon and Laitin (2003)", height=2.5, col0="white", col1="black")

separationplot(pred.CH.war, Warstds, type = "line", line = T, lwd2=1, show.expected=T,
heading="Collier and Hoeffler (2004)", height=2.5, col0="white", col1="black")

separationplot(pred.HR.war, Warstds, type = "line", line = T, lwd2=1, show.expected=T,
heading="Hegre and Sambanis (2006)", height=2.5, col0="white", col1="black")
```

```
### Plot Partial Dependence Plots###
#########################################################################
#####################################
# par(mfrow=c(3,3))
# partialPlot(RF.out.am, data2, gdpgrowth, which.class="1", xlab="GDP Growth Rate",
main="",
# ylab=expression(paste(Delta, "Fraction of Votes p(Y=1)")))
# partialPlot(RF.out.am, data2, ln_gdpen, ylim=c(-0.15, 0.15), which.class="1",
xlab="GDP per Capita (log)",
# main="", ylab=expression(paste(Delta, "Fraction of Votes p(Y=1)")))
# partialPlot(RF.out.am, data2, life, ylim=c(-0.15, 0.15), which.class="1", xlab="Life
Expectancy",
# main="", ylab=expression(paste(Delta, "Fraction of Votes p(Y=1)")))
# partialPlot(RF.out.am, data2, infant, ylim=c(-0.15, 0.15), which.class="1", xlab="Infant
Mortality Rate",
# main="", ylab=expression(paste(Delta, "Fraction of Votes p(Y=1)")))
# partialPlot(RF.out.am, data2, lmtnest, ylim=c(-0.15, 0.15), which.class="1",
xlab="Mountainous Terrain (log)"
#, main="",  ylab=expression(paste(Delta, "Fraction of Votes p(Y=1)")))
# partialPlot(RF.out.am, data2, pol4sq, ylim=c(-0.15, 0.15), which.class="1", xlab="Polity
IV Sq",
# main="", ylab=expression(paste(Delta, "Fraction of Votes p(Y=1)")))
# partialPlot(RF.out.am, data2, lpopns, ylim=c(-0.15, 0.15), which.class="1",
xlab="Population", main="",
# ylab=expression(paste(Delta, "Fraction of Votes p(Y=1)")))
# partialPlot(RF.out.am, data2, trade, ylim=c(-0.15, 0.15), which.class="1", xlab="Trade",
main="",
# xlim=c(0,200), ylab=expression(paste(Delta, "Fraction of Votes p(Y=1)")))
# partialPlot(RF.out.am, data2, geo1, ylim=c(-0.15, 0.15), which.class="1", xlab="W.
Europe and U.S.",
# main="",ylab=expression(paste(Delta, "Fraction of Votes p(Y=1)")))
#########################################################################
#####################################



# This section provides the correct code for the imputing of the out-of-sample data
# to replicate the new and corrected Table 1.
# Originally this section of code was uploaded in error, along with an incorrect dataset,
# see our response in the forum.
# This corrected code provides the means to impute the out-of-sample data and
generate
# the out-of-sample predictions as per our response.
# The substance of our results do not change, though we are unable to replicate Table 1
# exactly due to loss of original code and data.
```

```
# Random Forest predicts more CW onsets in out-of-sample data than logistic
regression.


# Seed for Imputation of out-of-sample data.
set.seed(425)


### Dataset for imputation.
data_imp<-read.csv(file="data_full.csv")

# Imputation procedure.
# This is the imputation procedure we originally used to impute this data.

rf.imp<-rfImpute(data_imp, as.factor(data_imp$warstds), iter=5, ntree=1000)


###Out of Sample Data ###

# Subsetting imputed data.
mena<-subset(rf.imp, rf.imp$year > 2000)



### Generate out of sample predictions for Table 1 (corrected)


fl.pred<-predict(model.fl.1, newdata=mena, type="prob")
fl.pred<-as.data.frame(fl.pred)
pred.FL.1<-prediction(fl.pred$war, mena$`as.factor(data_imp$warstds)`)
perf.FL.1<-performance(pred.FL.1, "auc")

ch.pred<-predict(model.ch.1, newdata=mena, type="prob")
ch.pred<-as.data.frame(ch.pred)
pred.CH.1<-prediction(ch.pred$war, mena$`as.factor(data_imp$warstds)`)
perf.CH.1<-performance(pred.CH.1, "auc")

hs.pred<-predict(model.hs.1, newdata=mena, type="prob")
hs.pred<-as.data.frame(hs.pred)
pred.HS.1<-prediction(hs.pred$war, mena$`as.factor(data_imp$warstds)`)
perf.HS.1<-performance(pred.HS.1, "auc")


rf.pred<-predict(model.rf, newdata=mena, type="prob")
rf.pred<-as.data.frame(rf.pred)

pred.RF.1<-prediction(rf.pred$war, mena$`as.factor(data_imp$warstds)`)
```

```
perf.RF.1<-performance(pred.RF.1, "tpr", "fpr")
perf.RF.1<-performance(pred.RF.1, "auc")

### Save Imputed Data. ###

predictions<-cbind(mena$cowcode, mena$year, mena$warstds, fl.pred[,2], ch.pred[,2],
hs.pred[,2], rf.pred[,2])

### Write column headings for the out of sample data. ###
colnames(predictions)<-c("COWcode", "year", "CW_Onset", "Fearon and Latin (2003)",
"Collier and Hoeffler (2004)", "Hegre and Sambanis (2006)",
"Random Forest")

### Save predictions as data frame for ordering the columns.
predictions<-as.data.frame(predictions)

### Table 1 Results, ordered by Onset (decreasing), and year (increasing) in R rather
than excel.
Onset_table<-predictions[order(-predictions$CW_Onset, predictions$year),]

### Rows 1-19 of the above go in Table 1. ###
Onset_table_1thru19<-head(Onset_table, n=19)

### Here's the code for Table 1 in Latex. ###
xtable(Onset_table_1thru19)

### Write the .csv file for all predictions to check against the Latex code for Table 1.
### Sort the csv same way as the Latex table - CW_Onset (decreasing), then by year
(increasing).

write.csv(predictions, file="testing_file_CW_onset_RF_rep_final.csv")
```