Final Project

Samira Zarandioon and Bradley Robinson

8/10/2018

Introduction

Data Description

Exploratory Analysis

Addressing Objective 1:

```
Restatement of Problem and the overall approach to solve it Required
```

Model Selection Required

Type of Selection

```
Any or all:     , RIDGE, ELASTIC NET,
Stepwise, Forward, Backward
Manual / Intuition
```

Checking Assumptions Required

```
                                Lack of fit test
                                Influential point analysis (Cook's D and Leverage)
Optional  Residual Plots
```

Parameter Interpretation

```
Interpretation  Required
Confidence Intervals Required
```

Final conclusions from the analyses of Objective 1 Required

Addressing Objective 2

Make sure it is clear how many models were created to compete against the one in Objective 1. Make note of any tuning parameters that were used and how you came up with them (knn and random forest logistics) Required

Main Analysis Content Required Overall report of the error metrics on a test set or CV run. Also if the two best models have error rates of .05 and .045, can we really say that one model is outperforming the other? What other tools that we learned in the second half of this class that could help us get at that?

Conclusion/Discussion Required

The conclusion should reprise the questions and conclusions of objective 2 with recommendations of

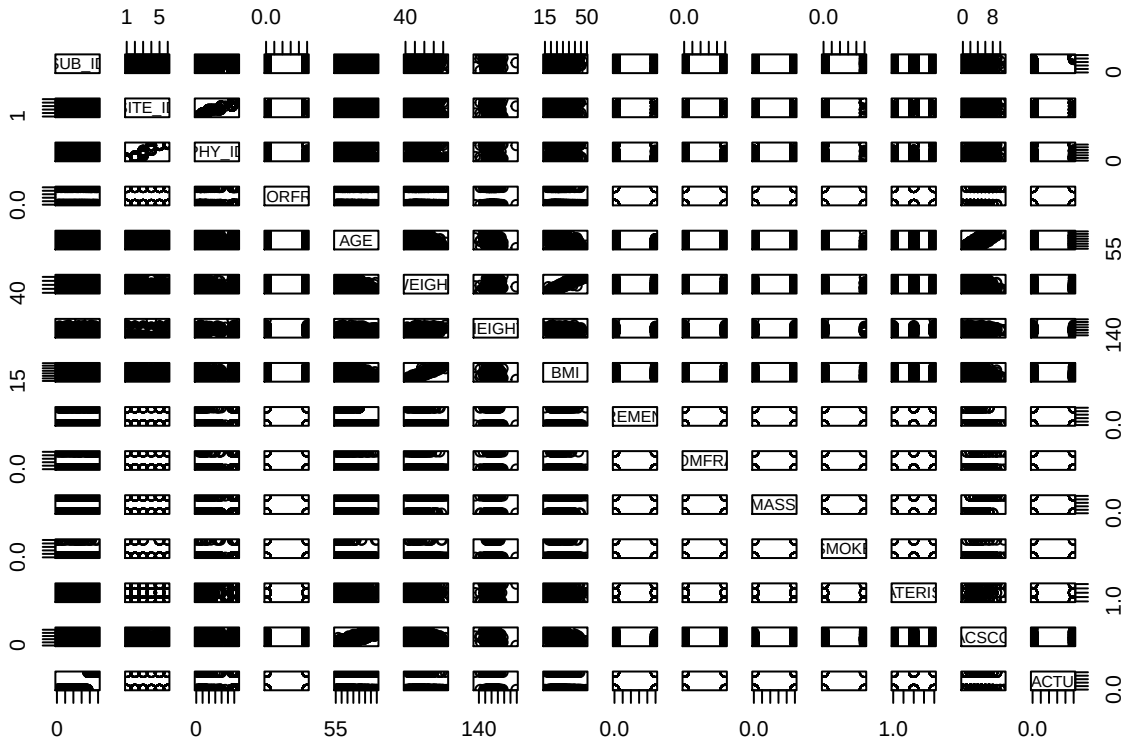
Appendix Required

Well commented SAS/R Code Required
Graphics and summary tables (Can be placed in the appendix or in the written report itself.)

```
library(MASS)       # provides LDA & QDA model functions


glow500 <- read.csv(file="glow500.csv",head=TRUE,sep=",")

set.seed(123)
sample <- sample(c(TRUE, FALSE), nrow(glow500), replace = T, prob = c(0.6,0.4))
train <- glow500[sample, ]
test <- glow500[!sample, ]

(lda.m1 <- lda(FRACTURE ~ RATERISK + FRACSCORE + HEIGHT, data = train))
```

```
## Call:
## lda(FRACTURE ~ RATERISK + FRACSCORE + HEIGHT, data = train)
##
## Prior probabilities of groups:
##         0         1
## 0.7508197 0.2491803
##
## Group means:
##   RATERISK FRACSCORE   HEIGHT
## 0 1.882096  3.410480 161.9389
## 1 2.118421  4.815789 160.0789
##
## Coefficients of linear discriminants:
##                   LD1
## RATERISK   0.5461345
## FRACSCORE  0.3285060
## HEIGHT    -0.0516470
```

```
pairs(glow500)
```




```
#The cor() function produces a matrix that contains all of the pairwise correlations among the predicto
cor(glow500)
```

```
##                SUB_ID     SITE_ID      PHY_ID    PRIORFRAC         AGE
## SUB_ID     1.00000000  0.04844619  0.04469372  0.179556091  0.11702656
## SITE_ID    0.04844619  1.00000000  0.97516015 -0.022477436  0.02644718
## PHY_ID     0.04469372  0.97516015  1.00000000 -0.010830605  0.02271392
## PRIORFRAC  0.17955609 -0.02247744 -0.01083061  1.000000000  0.29145290
## AGE        0.11702656  0.02644718  0.02271392  0.291452898  1.00000000
## WEIGHT    -0.01351774 -0.05952453 -0.06222351 -0.023993946 -0.27159637
## HEIGHT    -0.05944021 -0.13009606 -0.14411659 -0.102203188 -0.19264861
## BMI        0.01157865 -0.01431135 -0.01286993  0.003314938 -0.22125651
## PREMENO   -0.02486197 -0.05604168 -0.05414796  0.006477171 -0.15911055
## MOMFRAC    0.11991894  0.05410014  0.05338689  0.022190273  0.03474619
## ARMASSIST  0.11717237  0.05643701  0.04206567  0.196139746  0.23831932
## SMOKE     -0.03505559  0.01600795  0.02312727  0.057413702 -0.09048779
## RATERISK   0.11323889  0.08516374  0.08315208  0.174844904 -0.04889352
## FRACSCORE  0.17637715  0.06300991  0.05670781  0.486079265  0.86991650
## FRACTURE   0.75000150  0.06935643  0.06745920  0.218088192  0.20765352
##                 WEIGHT       HEIGHT          BMI      PREMENO     MOMFRAC
## SUB_ID    -0.013517738 -0.059440206  0.011578645 -0.024861968  0.11991894
## SITE_ID   -0.059524527 -0.130096059 -0.014311354 -0.056041680  0.05410014
## PHY_ID    -0.062223509 -0.144116593 -0.012869934 -0.054147961  0.05338689
## PRIORFRAC -0.023993946 -0.102203188  0.003314938  0.006477171  0.02219027
## AGE       -0.271596372 -0.192648608 -0.221256511 -0.159110550  0.03474619
## WEIGHT     1.000000000  0.315969149  0.937336030  0.080381676 -0.06124937
## HEIGHT     0.315969149  1.000000000 -0.024376893 -0.009008094  0.06963166
## BMI        0.937336030 -0.024376893  1.000000000  0.094600200 -0.08804359
## PREMENO    0.080381676 -0.009008094  0.094600200  1.000000000 -0.00917403
## MOMFRAC   -0.061249375  0.069631660 -0.088043587 -0.009174030  1.00000000
## ARMASSIST  0.319197889  0.070604294  0.308034603  0.078605874  0.00687544
```

```
## SMOKE      0.002906384 -0.024370933  0.008832892  0.103278242 -0.01281949
## RATERISK  -0.082881514 -0.016604810 -0.084304949  0.075919297  0.12473287
## FRACSCORE -0.161375361 -0.161995249 -0.120347231 -0.078528348  0.17564672
## FRACTURE  -0.036259440 -0.136400553  0.014985061  0.008760366  0.10643875
##             ARMASSIST        SMOKE     RATERISK   FRACSCORE     FRACTURE
## SUB_ID     0.11717237 -0.035055586  0.113238894  0.17637715  0.750001500
## SITE_ID    0.05643701  0.016007953  0.085163740  0.06300991  0.069356431
## PHY_ID     0.04206567  0.023127267  0.083152085  0.05670781  0.067459202
## PRIORFRAC  0.19613975  0.057413702  0.174844904  0.48607927  0.218088192
## AGE        0.23831932 -0.090487793 -0.048893522  0.86991650  0.207653516
## WEIGHT     0.31919789  0.002906384 -0.082881514 -0.16137536 -0.036259440
## HEIGHT     0.07060429 -0.024370933 -0.016604810 -0.16199525 -0.136400553
## BMI        0.30803460  0.008832892 -0.084304949 -0.12034723  0.014985061
## PREMENO    0.07860587  0.103278242  0.075919297 -0.07852835  0.008760366
## MOMFRAC    0.00687544 -0.012819494  0.124732868  0.17564672  0.106438749
## ARMASSIST  1.00000000  0.062141897  0.122703011  0.57269737  0.152567885
## SMOKE      0.06214190  1.000000000  0.003961627  0.07725582 -0.031679398
## RATERISK   0.12270301  0.003961627  1.000000000  0.08206561  0.151731878
## FRACSCORE  0.57269737  0.077255822  0.082065606  1.00000000  0.264479514
## FRACTURE   0.15256788 -0.031679398  0.151731878  0.26447951  1.000000000
```

Train randomforest:

```
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

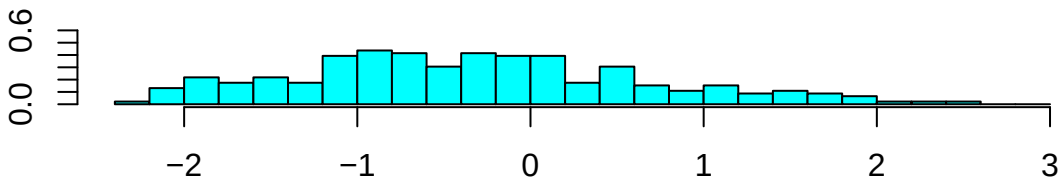
```
train$FRACTURE_F <- as.factor(train$FRACTURE)
rf.m <- randomForest(FRACTURE_F ~ RATERISK + FRACSCORE + HEIGHT, data=train, maxnodes=4, ntree=30)
test.predicted.rf <- predict(rf.m, newdata = test, type="response")
```

Linear Discriminant Analysis

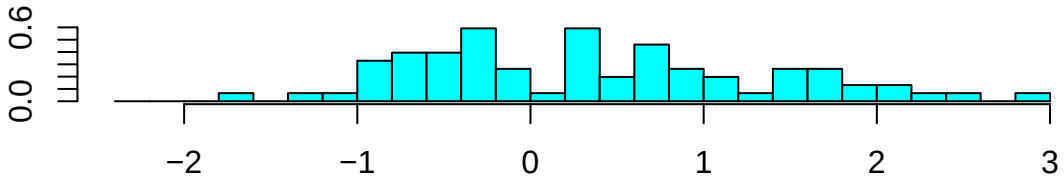
```
(lda.m1 <- lda(FRACTURE ~ RATERISK + FRACSCORE + HEIGHT, data = train))
```

```
## Call:
## lda(FRACTURE ~ RATERISK + FRACSCORE + HEIGHT, data = train)
##
## Prior probabilities of groups:
##         0         1
## 0.7508197 0.2491803
##
## Group means:
##   RATERISK FRACSCORE   HEIGHT
## 0 1.882096  3.410480 161.9389
## 1 2.118421  4.815789 160.0789
##
## Coefficients of linear discriminants:
##                  LD1
## RATERISK   0.5461345
## FRACSCORE  0.3285060
## HEIGHT    -0.0516470
```

```
plot(lda.m1)
```



group 0



group 1

```
(qda.m1 <- qda(FRACTURE ~ RATERISK + FRACSCORE + HEIGHT, data = train))
```

```
## Call:
## qda(FRACTURE ~ RATERISK + FRACSCORE + HEIGHT, data = train)
##
## Prior probabilities of groups:
##         0         1
## 0.7508197 0.2491803
##
## Group means:
##   RATERISK FRACSCORE   HEIGHT
## 0 1.882096  3.410480 161.9389
## 1 2.118421  4.815789 160.0789
```

Train linear regression model:

```
(glm.fit <- glm(FRACTURE ~ RATERISK + FRACSCORE + HEIGHT, data = train))
```

```
##
## Call:  glm(formula = FRACTURE ~ RATERISK + FRACSCORE + HEIGHT, data = train)
##
## Coefficients:
## (Intercept)     RATERISK    FRACSCORE       HEIGHT
##    0.967155     0.064873     0.039022    -0.006135
##
## Degrees of Freedom: 304 Total (i.e. Null);  301 Residual
## Null Deviance:       57.06
## Residual Deviance: 52.41     AIC: 338.4
```

Evaluate random forest:

```
library(randomForest)
train$FRACTURE <- as.factor(train$FRACTURE)
rf.m <- randomForest(FRACTURE ~ RATERISK + FRACSCORE + HEIGHT, data=train, maxnodes=5, ntree=5)
```

```r
test.predicted.rf <- predict(rf.m, newdata = test, type="response")

test.predicted.rf <- predict(rf.m, newdata = test, type="response")
# confusion matrix
table(test$FRACTURE, test.predicted.rf)
```

```
##    test.predicted.rf
##       0   1
##   0 146   0
##   1  49   0
```

```r
# accuracy rate
mean(test.predicted.rf == test$FRACTURE)
```

```
## [1] 0.7487179
```

Evaluate Linear Discriminant Analysis Model:

```r
# predictions
test.predicted.lda <- predict(lda.m1, newdata = test)

# confusion matrix
table(test$FRACTURE, test.predicted.lda$class)
```

```
##
##       0   1
##   0 140   6
##   1  41   8
```

```r
# accuracy rate
mean(test.predicted.lda$class == test$FRACTURE)
```

```
## [1] 0.7589744
```

```r
# predictions
test.predicted.qda <- predict(qda.m1, newdata = test)

# confusion matrix
table(test$FRACTURE, test.predicted.qda$class)
```

```
##
##       0   1
##   0 140   6
##   1  45   4
```

```r
# accuracy rate
mean(test.predicted.qda$class == test$FRACTURE)
```

```
## [1] 0.7384615
```

Evaluate Linear Regression Model:

Confusion Metrix:

```r
# predictions
glm.probs <- predict(glm.fit, test, type = "response")

# confusion matrix
table(test$FRACTURE, ifelse(glm.probs < 0.5, 0, 1))
```

```
##
##       0   1
##   0 142   4
##   1  47   2
```

Accuracy:

```
# accuracy rate
mean(ifelse(glm.probs > 0.5, 1 , 0) == test$FRACTURE)
```

```
## [1] 0.7384615
```

```
# ROC curves
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:randomForest':
##
##     combine
```

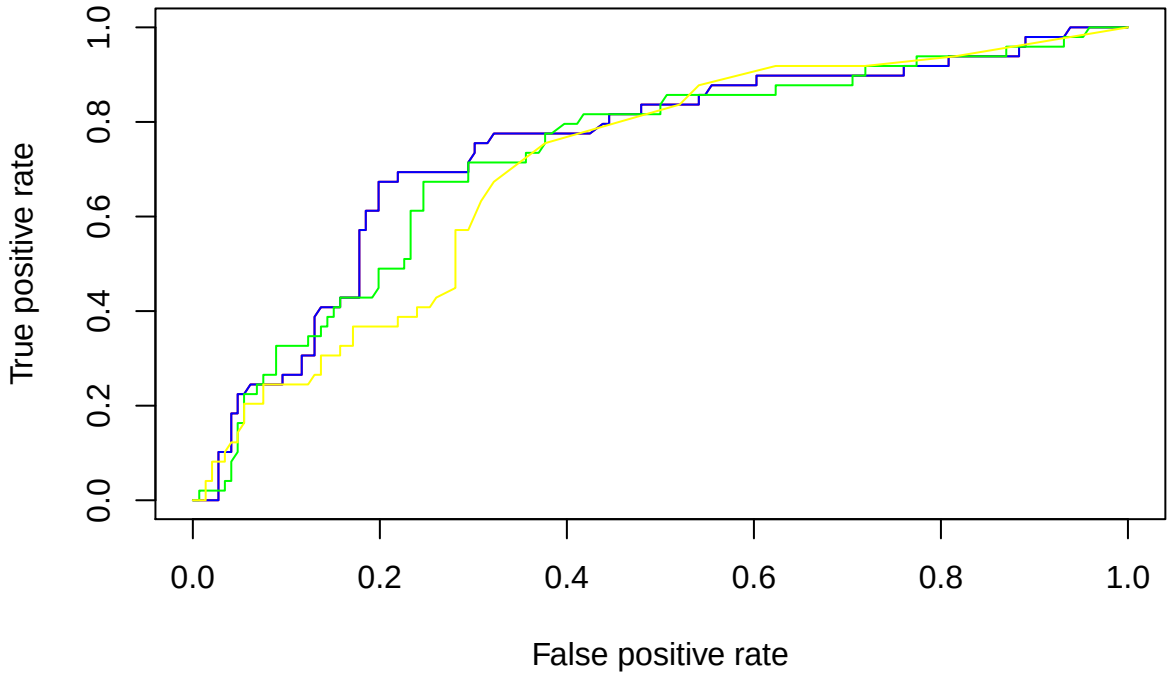
```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(randomForest)
train$FRACTURE <- as.factor(train$FRACTURE)
rf.m <- randomForest(FRACTURE ~ RATERISK + FRACSCORE + HEIGHT, data=train, maxnodes=5, ntree= 1000)
test.predicted.rf <- predict(rf.m, newdata = test, type="response")


logistic <- prediction(glm.probs, test$FRACTURE) %>%
  performance(measure = "tpr", x.measure = "fpr")

lda <- prediction(test.predicted.lda$posterior[,2], test$FRACTURE) %>%
  performance(measure = "tpr", x.measure = "fpr")

qda <- prediction(test.predicted.qda$posterior[,2], test$FRACTURE) %>%
  performance(measure = "tpr", x.measure = "fpr")
```

```
test.predicted.rf.prob <- predict(rf.m, newdata = test, type="prob")
rf <- prediction(test.predicted.rf.prob[,2], test$FRACTURE) %>%
  performance(measure = "tpr", x.measure = "fpr")

plot(logistic, col = "red")
plot(lda, add = TRUE, col = "blue")
plot(qda, add = TRUE, col = "green")
plot(rf, add = TRUE, col = "yellow")
```




```
# Logistic regression AUC
prediction(glm.probs, test$FRACTURE) %>%
  performance(measure = "auc") %>%
  .@y.values
```

```
## [[1]]
## [1] 0.7435001
```

```
# LDA AUC
prediction(test.predicted.lda$posterior[,2], test$FRACTURE) %>%
  performance(measure = "auc") %>%
  .@y.values
```

```
## [[1]]
## [1] 0.7435001
```

```
# QDA AUC
prediction(test.predicted.qda$posterior[,2], test$FRACTURE) %>%
  performance(measure = "auc") %>%
  .@y.values
```

```
## [[1]]
## [1] 0.7264467
```

```r
# RandomForest AUC
prediction(test.predicted.rf.prob[,2], test$FRACTURE) %>%
  performance(measure = "auc") %>%
  .@y.values
```

```
## [[1]]
## [1] 0.7038021
```

SAS Codes:

```
LIBNAME MYSASLIB '/home/szarandioon0/';
DATA GLOW500_ORIG;
INFILE '/home/szarandioon0/statistics2/Project2/glow500.csv' DLM = ',' FIRSTOBS = 2;
INPUT SUB_ID SITE_ID PHY_ID PRIORFRAC AGE WEIGHT HEIGHT BMI PREMENO MOMFRAC ARMASSIST SMOKE RATERISK FRA
RUN;

DATA GLOW500(DROP = SUB_ID);
SET GLOW500_ORIG;
RUN;

proc factor data=GLOW500 simple corr;
run;

ods graphics on;
proc princomp data=GLOW500 plots(ncomp=3)=all n=5;
run;

proc candisc data=GLOW500 out=discrim_out ;
  class FRACTURE;
  var SITE_ID   PHY_ID PRIORFRAC AGE WEIGHT HEIGHT BMI PREMENO MOMFRAC ARMASSIST SMOKE RATERISK FRACSCOR
run;

    title 'Stepwise Regression on Global Longitudinal Study of Osteoporosis in Women (GLOW) Dataset';
    proc logistic data=GLOW500 outest=betas covout;
    model FRACTURE(event='1')=SITE_ID PHY_ID PRIORFRAC AGE WEIGHT HEIGHT BMI PREMENO    MOMFRAC ARMASSIS
                        / selection=stepwise;
    output out=pred p=phat lower=lcl upper=ucl predprob=(individual crossvalidate);
    run;

data train test;
set GLOW500;
if rand('uniform') <= 0.3
then output test;
else output train;
run;

ods graphics on;
proc logistic data=train;
model FRACTURE(event="1") = RATERISK FRACSCORE HEIGHT / outroc=troc;
score data=test out=valpred outroc=vroc;
roc; roccontrast;
run;

proc logistic data=train plots(only)=roc;
model FRACTURE(event="1") = RATERISK FRACSCORE HEIGHT;
```

```
run;

proc logistic data=train rocoptions(crossvalidate) plots(only)=roc;
model FRACTURE(event="1") = RATERISK FRACSCORE HEIGHT;
run;

proc discrim data=train testdata=test canonical;
class FRACTURE;
var SITE_ID PHY_ID PRIORFRAC AGE WEIGHT HEIGHT BMI PREMENO MOMFRAC ARMASSIST SMOKE RATERISK FRACSCORE;
run;

proc hpforest data=train;
target FRACTURE/level=nominal;
input PRIORFRAC PREMENO MOMFRAC ARMASSIST SMOKE/level=nominal;
input SITE_ID PHY_ID AGE WEIGHT HEIGHT BMI RATERISK FRACSCORE/level=interval;
run;
```

Including Plots

You can also embed plots, for example: