Template for Assignment 7— Predicting match outcome

Student Name

- **Pranav Vinod**

Notes:

- ***First, please re-save this document on your computer, RENAMING the file to contain your last name.***
- ***Point values of each part are shown below; 10 points will be allocated for the quality of your business writing (organization, clarity, grammar, etc.).***
- ***Type or paste your responses into the boxes below. The boxes will expand to fit your answers.***
- ***Deliverables- Upload following 2 files on the course website:***
  1) This completed file.
  2) R file used.

Classification and Prediction Models

0. Read the betting.csv data file into RStudio. Run set.seed(XXX) by using last three digits of your student ID in place of XXX followed by partitioning of the dataset into training (50%) and testing (50%). Report on how many cases of Win, Draw, and loss exist in the training and testing data. (10 pts).

```
set.seed(464)
ind <- sample(2, nrow(mydata), replace = T, prob = c(0.5, 0.5))
train <- mydata[ind == 1,]
test <- mydata[ind == 2,]
```

```
> table(train$Match0)

Draw Loss  Win
 216  215  349
> table(test$Match0)

Draw Loss  Win
 188  177  375
```

There are 216, 215 and 349 cases of Draw, Loss and Win respectively in the Train dataset.

There are 188, 177 and 375 cases of Draw, Loss and Win respectively in the Test dataset.

1. Develop a multinomial logistic regression model for predicting match outcome based on the training dataset and report the final model, prediction equations, confusion matrix for training and test datasets. What conclusions can you derive regarding betting about match outcome at game half-time? (20 points):

```
> summary(m)
Call:
multinom(formula = Match0 ~ ., data = train)

Coefficients:
     (Intercept)  Match.Number       HTGD          REDH       REDA      POINTSH      POINTSA
Loss  -3.8223512 -2.462056e-04 -0.7633176  6.883169e-05 -0.6406233 -0.003648928  0.003085560
Win   -0.6494705  2.217083e-05  1.4193793 -1.188034e+00  0.2389115  0.003938018 -0.002660112
          TOTALHP     TOTALAP     FGS01      FGS11
Loss -0.007121153  0.01108467 4.2028080 3.45810612
Win   0.013590752 -0.00641993 0.2397604 0.04155586

Std. Errors:
     (Intercept) Match.Number      HTGD      REDH      REDA    POINTSH    POINTSA     TOTALHP
Loss   0.2775831 0.0002707577 0.1802526 0.4066500 0.4791739 0.01260996 0.01155423 0.005241029
Win    0.3575379 0.0002313775 0.2105292 0.1054282 0.6395716 0.01034633 0.01076946 0.004645095
        TOTALAP     FGS01     FGS11
Loss 0.005413193 0.2135065 0.2298107
Win  0.004456842 0.2753747 0.2901700

Residual Deviance: 1117.186
AIC: 1161.186
```

The prediction equations are as follows:

$$\log\left(\frac{p(Loss)}{p(Draw)}\right) = -3.8 - 0.76 * HTGD + 0.00006 * REDH - 0.64 * REDA - 0.003 * POINTSH + 0.003 * POINTSA - 0.007 * TOTALHP + 0.011 * TOTALAP + 4.20 * FGS0 + 3.45 * FGS1$$

$$\log\left(\frac{p(Win)}{p(Draw)}\right) = -0.649 + 1.419 * HTGD - 1.188 * REDH + 0.239 * REDA + 0.004 * POINTSH - 0.002 * POINTSA + 0.013 * TOTALHP - 0.006 * TOTALAP + 0.239 * FGS0 + 0.041 * FGS1$$

Confusion Matrix for Train

```
Confusion Matrix and Statistics

          Reference
Prediction Draw Loss Win
      Draw   80   10  53
      Loss   63  178  35
      Win    73   27 261

Overall Statistics

               Accuracy : 0.6654
                 95% CI : (0.6311, 0.6985)
    No Information Rate : 0.4474
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4809

 Mcnemar's Test P-Value : 2.869e-09

Statistics by Class:

                     Class: Draw Class: Loss Class: Win
Sensitivity               0.3704      0.8279     0.7479
Specificity               0.8883      0.8265     0.7680
Pos Pred Value            0.5594      0.6449     0.7230
Neg Pred Value            0.7865      0.9266     0.7900
Prevalence                0.2769      0.2756     0.4474
Detection Rate            0.1026      0.2282     0.3346
Detection Prevalence      0.1833      0.3538     0.4628
Balanced Accuracy         0.6293      0.8272     0.7579
```

Confusion Matrix for test data

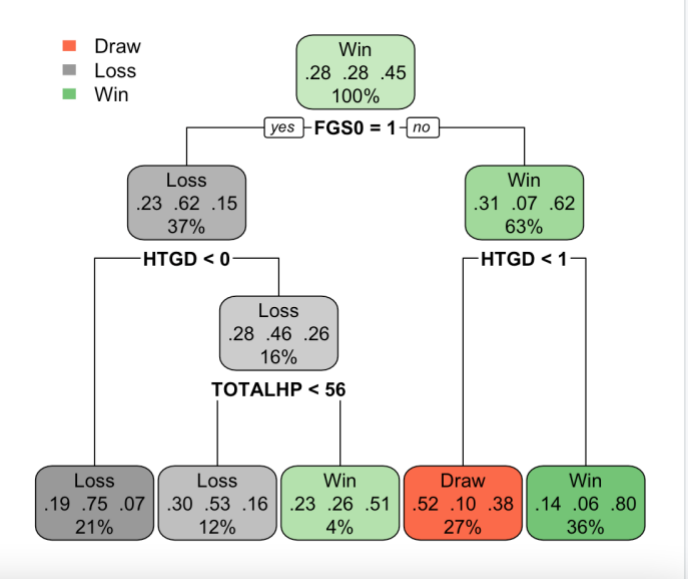
```
Confusion Matrix and Statistics

          Reference
Prediction Draw Loss Win
      Draw   64   11  63
      Loss   62  145  44
      Win    62   21 268

Overall Statistics

               Accuracy : 0.6446
                 95% CI : (0.6089, 0.6791)
    No Information Rate : 0.5068
    P-Value [Acc > NIR] : 2.742e-14

                  Kappa : 0.4369

 Mcnemar's Test P-Value : 1.683e-09

Statistics by Class:

                     Class: Draw Class: Loss Class: Win
Sensitivity              0.34043      0.8192     0.7147
Specificity              0.86594      0.8117     0.7726
Pos Pred Value           0.46377      0.5777     0.7635
Neg Pred Value           0.79402      0.9346     0.7249
Prevalence               0.25405      0.2392     0.5068
Detection Rate           0.08649      0.1959     0.3622
Detection Prevalence     0.18649      0.3392     0.4743
Balanced Accuracy        0.60318      0.8155     0.7436
```

From the confusion matrices, we can see that the overall accuracy is 66% for the training set and 64% for the testing set. Among the three different classes, the algorithm is best at correctly predicting “Loss” for both the

train and test set with sensitivity of 0.83 and 0.72 respectively. It is the worst at correctly predicting "Draw" for both the sets.

2. Develop a decision tree for predicting match outcome using training dataset and report the final tree, related code, confusion matrix for both training and test datasets. What conclusions can you derive (20 points):

```
# Tree
tree <- rpart(Match0 ~., data = train)
rpart.plot(tree)
printcp(tree)
plotcp(tree)

# Confusion matrix -train
p <- predict(tree, train, type = 'class')
confusionMatrix(p, train$Match0)

# Confusion matrix -test
p <- predict(tree, test, type = 'class')
confusionMatrix(p, test$Match0)


# ROC
p1 <- predict(tree, test, type = 'prob')
p1 <- p1[,2]
r <- multiclass.roc(test$Match0, p1, percent = TRUE)
roc <- r[['rocs']]
r1 <- roc[[1]]
plot.roc(r1,
         print.auc=TRUE,
         auc.polygon=TRUE,
         grid=c(0.1, 0.2),
         grid.col=c("green", "red"),
         max.auc.polygon=TRUE,
         auc.polygon.col="lightblue",
         print.thres=TRUE,
         main= 'ROC Curve')
```



From the final tree, we can see that FGS0, the first goal scored by the away team is the most important factor in deciding how the game ends. Depending on whether the first goal was scored by the away team or not, the next most important attribute is the half time goal difference, HTGD.

If the first goal was not scored by the away team, then HTGD is the only other factor which decides the outcome of the match. If it is less than 0 then the match likely ends in a draw and if it is greater than 1, the home team usually wins the game.

If the first goal was scored by the away team, then along with HTGD, TOTALHP is also a factor in predicting the outcome.

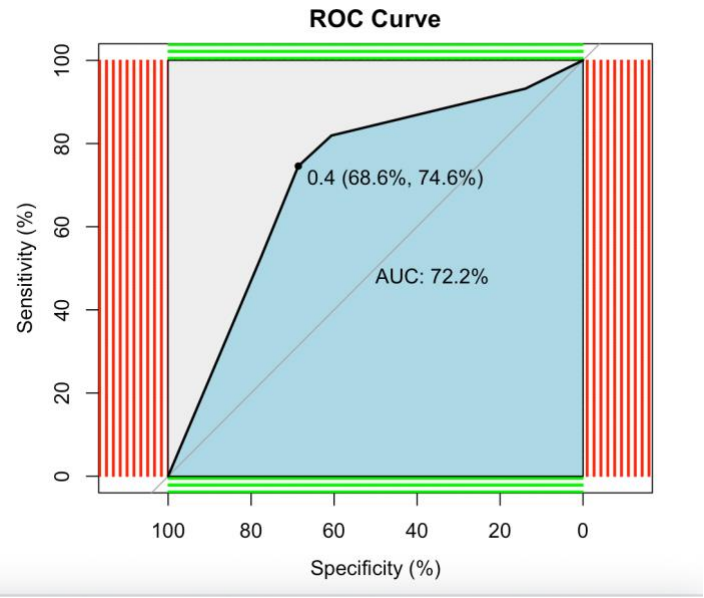
Confusion Matrix for Train set

```
Confusion Matrix and Statistics

          Reference
Prediction Draw Loss Win
      Draw  110   20  80
      Loss   58  170  26
      Win    48   25 243

Overall Statistics

               Accuracy : 0.6705
                 95% CI : (0.6363, 0.7034)
    No Information Rate : 0.4474
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4965

 Mcnemar's Test P-Value : 7.378e-06

Statistics by Class:

                     Class: Draw Class: Loss Class: Win
Sensitivity               0.5093      0.7907     0.6963
Specificity               0.8227      0.8513     0.8306
Pos Pred Value            0.5238      0.6693     0.7690
Neg Pred Value            0.8140      0.9144     0.7716
Prevalence                0.2769      0.2756     0.4474
Detection Rate            0.1410      0.2179     0.3115
Detection Prevalence      0.2692      0.3256     0.4051
Balanced Accuracy         0.6660      0.8210     0.7635
```

Confusion Matrix for Test set

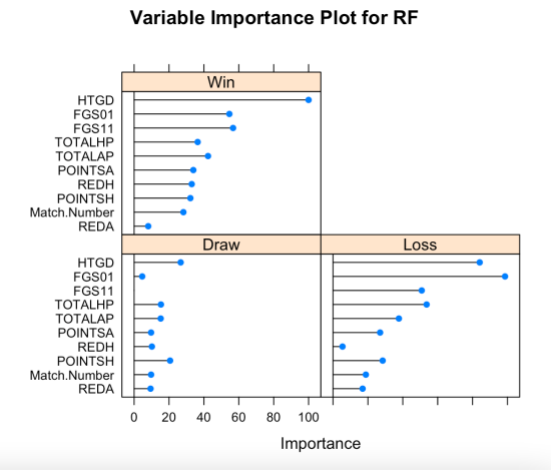
```
Confusion Matrix and Statistics

          Reference
Prediction Draw Loss Win
      Draw   88   20 103
      Loss   59  132  48
      Win    41   25 224

Overall Statistics

               Accuracy : 0.6
                 95% CI : (0.5637, 0.6355)
    No Information Rate : 0.5068
    P-Value [Acc > NIR] : 2.154e-07

                  Kappa : 0.3862

 Mcnemar's Test P-Value : 1.667e-11

Statistics by Class:

                     Class: Draw Class: Loss Class: Win
Sensitivity               0.4681      0.7458     0.5973
Specificity               0.7772      0.8099     0.8192
Pos Pred Value            0.4171      0.5523     0.7724
Neg Pred Value            0.8110      0.9102     0.6644
Prevalence                0.2541      0.2392     0.5068
Detection Rate            0.1189      0.1784     0.3027
Detection Prevalence      0.2851      0.3230     0.3919
Balanced Accuracy         0.6226      0.7779     0.7083
```

From the confusion matrices, we can see that the overall accuracy is 67% for the training set and 60% for the testing set. Among the three different classes, the algorithm is best at correctly predicting “Loss” for both the train and test set with sensitivity of 0.79 and 0.74 respectively. It is the worst at correctly predicting “Draw” for both the sets.



The model has an area under the curve of 72.2% with the threshold point being (0.686, 0.746).

3. Develop a random forest model for predicting match outcome using training dataset and provide confusion matrix for both training and test datasets. What conclusions can you derive (20 points):



From the Variable Importance plot we can see for Win and Draw, HTGD is the most important variable followed by FGS1 for Win and POINTSH for Draw. For Loss, FSG0 is the most important variable followed by HTGD.

Training Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction Draw Loss Win
      Draw  163    2   7
      Loss   25  200  17
      Win    28   13 325

Overall Statistics

               Accuracy : 0.8821
                 95% CI : (0.8573, 0.9038)
    No Information Rate : 0.4474
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8167

 Mcnemar's Test P-Value : 3.679e-07

Statistics by Class:

                     Class: Draw Class: Loss Class: Win
Sensitivity               0.7546      0.9302     0.9312
Specificity               0.9840      0.9257     0.9049
Pos Pred Value            0.9477      0.8264     0.8880
Neg Pred Value            0.9128      0.9721     0.9420
Prevalence                0.2769      0.2756     0.4474
Detection Rate            0.2090      0.2564     0.4167
Detection Prevalence      0.2205      0.3103     0.4692
Balanced Accuracy         0.8693      0.9279     0.9181
```

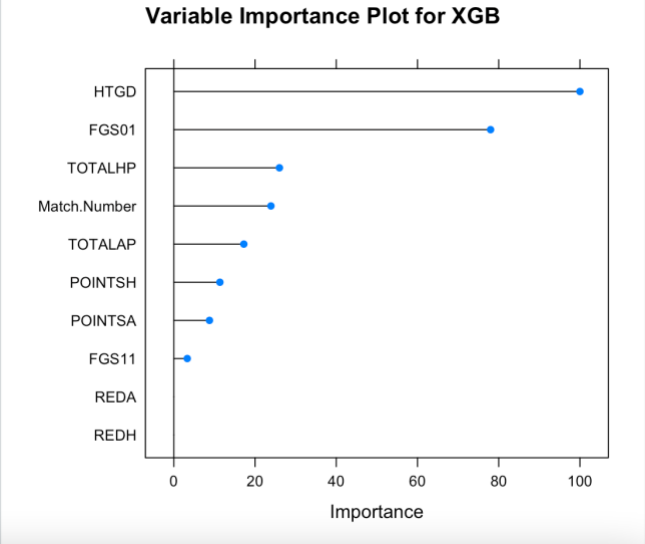
Testing Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction Draw Loss Win
      Draw   70   14  72
      Loss   68  144  50
      Win    50   19 253

Overall Statistics

               Accuracy : 0.6311
                 95% CI : (0.5952, 0.6659)
    No Information Rate : 0.5068
    P-Value [Acc > NIR] : 6.451e-12

                  Kappa : 0.4247

 Mcnemar's Test P-Value : 1.466e-11

Statistics by Class:

                     Class: Draw Class: Loss Class: Win
Sensitivity              0.37234      0.8136     0.6747
Specificity              0.84420      0.7904     0.8110
Pos Pred Value           0.44872      0.5496     0.7857
Neg Pred Value           0.79795      0.9310     0.7081
Prevalence               0.25405      0.2392     0.5068
Detection Rate           0.09459      0.1946     0.3419
Detection Prevalence     0.21081      0.3541     0.4351
Balanced Accuracy        0.60827      0.8020     0.7428
```

Further, from the confusion matrices we can see that the overall accuracy is 88% for the training set and 63% for the testing set. Among the three different classes, the algorithm is best at correctly predicting “Loss” for both the test set and “Win” for the train set with sensitivity of 0.81 and 0.93 respectively. It is the worst at

correctly predicting "Draw" for both the sets.

4. Develop an extreme gradient boosting model for predicting match outcome using training dataset and provide confusion matrix for both training and test datasets. What conclusions can you derive (20 points):



From the variable importance plot, HTGD is found to be the most important variable in predicting match outcome followed by FGS0 and TOTALHP.

Confusion Matrix for Training

```
> confusionMatrix(p, train$Match0)
Confusion Matrix and Statistics

          Reference
Prediction Draw Loss Win
      Draw  118   10  28
      Loss   45  187  25
      Win    53   18 296

Overall Statistics

               Accuracy : 0.7705
                 95% CI : (0.7394, 0.7996)
    No Information Rate : 0.4474
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6432

 Mcnemar's Test P-Value : 7.988e-07

Statistics by Class:

                     Class: Draw Class: Loss Class: Win
Sensitivity               0.5463      0.8698     0.8481
Specificity               0.9326      0.8761     0.8353
Pos Pred Value            0.7564      0.7276     0.8065
Neg Pred Value            0.8429      0.9465     0.8717
Prevalence                0.2769      0.2756     0.4474
Detection Rate            0.1513      0.2397     0.3795
Detection Prevalence      0.2000      0.3295     0.4705
Balanced Accuracy         0.7395      0.8729     0.8417
```

Confusion Matrix for Testing

```
> confusionMatrix(p, test$Match0)
Confusion Matrix and Statistics

          Reference
Prediction Draw Loss Win
      Draw   71   20  69
      Loss   56  139  43
      Win    61   18 263

Overall Statistics

               Accuracy : 0.6392
                 95% CI : (0.6034, 0.6739)
    No Information Rate : 0.5068
    P-Value [Acc > NIR] : 2.610e-13

                  Kappa : 0.4308

 Mcnemar's Test P-Value : 4.018e-06

Statistics by Class:

                     Class: Draw Class: Loss Class: Win
Sensitivity              0.37766      0.7853     0.7013
Specificity              0.83877      0.8242     0.7836
Pos Pred Value           0.44375      0.5840     0.7690
Neg Pred Value           0.79828      0.9243     0.7186
Prevalence               0.25405      0.2392     0.5068
Detection Rate           0.09595      0.1878     0.3554
Detection Prevalence     0.21622      0.3216     0.4622
Balanced Accuracy        0.60821      0.8047     0.7424
```

The parameters have been set to maximize testing accuracy with the lowest possible loss in accuracy and overfitting with the training set.

So, from the confusion matrices we can see that the overall accuracy is 77% for the training set and 64% for the testing set. Among the three different classes, the algorithm is best at correctly predicting "Loss" for both the test set and the train set with sensitivity of 0.78 and 0.87 respectively. It is the worst at correctly predicting "Draw" for both the sets.

5. Provide a summary of key results from the three models used above and compare results. Which classification and prediction method do you find to be the best for betting on the match outcome and why? (10 points):

	Train DT	Test DT	Train RF	Test RF	Train XGB	Test XGB
Accuracy (%)	67	60	88	63.1	77	64
Win Sensitivity	0.69	0.59	0.93	0.67	0.85	0.70
Loss Sensitivity	0.79	0.75	0.93	0.81	0.87	0.78
Draw Sensitivity	0.51	0.46	0.75	0.37	0.54	0.37

Considering the overall testing accuracy of each model, the Extreme Gradient Boosting method has the best performance. Among the three, the decision tree performs the worst.

At the same time, it must be noted that the random forest implementation is the best at correctly predicting if the match outcome will be "Loss" with the highest test sensitivity. So, if someone wants to bet that the home team would lose the match, they could consider using the random forest because of its high sensitivity.