**MATH 4432 Statistical Machine Learning Project 2**

**Classification of Football Betting Match Results II**

By WU, Sen (20271863) on 13 Apr. 2018

**Introduction**

This project is inspired by one of my best friends. He realized that it is possible to predict the handicap results of football matches by simply reading the odds and change of odds.

In fact, bookmakers over the world are always generating profits. The actuaries behind each bookmaker play crucial role in risk management. They make predictions of match results and suggest reasonable and attractive odds to boost the bets. They could make generous profits controlling the odds when some instances of major upsets happen.

In each football match, there are so many variables that could affect the match results. A lot of variables has been considered by actuaries before the match. Actuaries set the odds for making profit, thus we could do reverse engineering by odds from different bookmakers.

In project 1, we used techniques of linear regression and obtained a maximum accuracy of 56% on test data. With the non-linear fitting techniques introduced in chapters recently introduced, we may further increase the percentage of correct prediction. In this project, we will use two major approaches to predict the handicap results: One is the

**Data Set**

We will take the match results in past 5-6 years (From Season 2011-12 to 2016-17) from England Premier, German Division 1, Italian Division 1, French Division 1 and Spanish Division 1 as training data. There is a total of 9471 observations for training. We will take the match results of the above five leagues generated in current season (2017-18) as test data. As of 12 Apr. 2018, the number of training data is 1574.

The data source is the same to that in the first project. All data are obtainable from <http://www.football-data.co.uk/downloadm.php>. The detailed legend of data can be found at <http://www.football-data.co.uk/notes.txt>.

**Analysis**

After we update the test data set, we first import the .csv files it into R workspace:

> Football.train <- read.csv("C:/Users/.../Football\_data\_training.csv")

> Football.test <- read.csv("C:/Users/.../Football\_data\_test.csv")

> Football.test <- na.omit(Football.test)

> rownames(Football.test) <- seq(1:nrow(Football.test))

This time we will predict the handicap correct choice (CC) by two means. The first method is to predict the home-dividend (**h**an**d**i**c**ap **p**aying **o**dd of **h**ome team, HDCPOH) and away-dividend (**h**an**d**i**c**ap **p**aying **o**dd of **a**way team, HDCPOA) by using generalized additive models. After the home-dividend and away-dividend is computed by using the smoothing spline, we will compare the home and away-dividends to make the decision which team shall we invest.

> set.seed(1)

> gam1 <- gam(HDCPOH ~ s(B365H) + s(B365D) + s(B365A) + s(BWH) + s(BWD) + s(BWA) + s(IWH) + s(IWD) + s(IWA) + s(LBH) + s(LBD) + s(LBA) + s(PSH) + s(PSD) + s(PSA) + s(WHH) + s(WHD) + s(WHA) + s(VCH) + s(VCD) + s(VCA) + s(Bb1X2) + s(BbMxH) + s(BbAvH) + s(BbMxD) + s(BbAvD) + s(BbMxA) + s(BbAvA) + s(BbOU) + s(BbMx.2.5) + s(BbAv.2.5) + s(BbMx.2.5.1) + s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) + s(BbMxAHH) + s(BbAvAHH) + s(BbMxAHA) + s(BbAvAHA) + s(PSCH) + s(PSCD) + s(PSCA), data = Football.train)

> summary(gam1)

Call: gam(formula = HDCPOH ~ s(B365H) + s(B365D) + s(B365A) + s(BWH) +

s(BWD) + s(BWA) + s(IWH) + s(IWD) + s(IWA) + s(LBH) + s(LBD) +

s(LBA) + s(PSH) + s(PSD) + s(PSA) + s(WHH) + s(WHD) + s(WHA) +

s(VCH) + s(VCD) + s(VCA) + s(Bb1X2) + s(BbMxH) + s(BbAvH) +

s(BbMxD) + s(BbAvD) + s(BbMxA) + s(BbAvA) + s(BbOU) + s(BbMx.2.5) +

s(BbAv.2.5) + s(BbMx.2.5.1) + s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) +

s(BbMxAHH) + s(BbAvAHH) + s(BbMxAHA) + s(BbAvAHA) + s(PSCH) +

s(PSCD) + s(PSCA), data = Football.train)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.45733 -0.94060 0.02806 0.89072 1.70503

(Dispersion Parameter for gaussian family taken to be 0.7742)

Null Deviance: 7028.855 on 9039 degrees of freedom

Residual Deviance: 6867.998 on 8870.998 degrees of freedom

AIC: 23510.34

431 observations deleted due to missingness

Number of Local Scoring Iterations: 23

Anova for Parametric Effects

Df Sum Sq Mean Sq F value Pr(>F)

s(B365H) 1 1.0 0.963 1.2444 0.2646541

s(B365D) 1 9.3 9.284 11.9919 0.0005368 \*\*\*

s(B365A) 1 7.0 7.024 9.0724 0.0026023 \*\*

s(BWH) 1 0.1 0.056 0.0723 0.7880125

s(BWD) 1 0.1 0.108 0.1400 0.7082660

s(BWA) 1 0.1 0.076 0.0979 0.7543726

s(IWH) 1 5.2 5.221 6.7440 0.0094217 \*\*

s(IWD) 1 3.9 3.937 5.0848 0.0241603 \*

s(IWA) 1 2.6 2.588 3.3433 0.0675126 .

s(LBH) 1 1.4 1.370 1.7695 0.1834769

s(LBD) 1 3.5 3.500 4.5205 0.0335184 \*

s(LBA) 1 0.0 0.002 0.0023 0.9617614

s(PSH) 1 5.8 5.837 7.5396 0.0060480 \*\*

s(PSD) 1 5.9 5.896 7.6150 0.0058004 \*\*

s(PSA) 1 0.0 0.003 0.0036 0.9520702

s(WHH) 1 4.0 3.956 5.1092 0.0238237 \*

s(WHD) 1 0.1 0.075 0.0972 0.7552626

s(WHA) 1 1.1 1.057 1.3652 0.2426698

s(VCH) 1 1.7 1.717 2.2176 0.1364842

s(VCD) 1 2.1 2.121 2.7402 0.0978885 .

s(VCA) 1 1.9 1.941 2.5067 0.1134006

s(Bb1X2) 1 0.6 0.625 0.8070 0.3690449

s(BbMxH) 1 5.7 5.708 7.3725 0.0066356 \*\*

s(BbAvH) 1 0.0 0.010 0.0125 0.9109859

s(BbMxD) 1 3.9 3.949 5.1010 0.0239358 \*

s(BbAvD) 1 0.8 0.779 1.0061 0.3158666

s(BbMxA) 1 0.3 0.312 0.4034 0.5253536

s(BbAvA) 1 0.2 0.192 0.2486 0.6180481

s(BbOU) 1 0.1 0.128 0.1655 0.6841364

s(BbMx.2.5) 1 0.1 0.110 0.1418 0.7064933

s(BbAv.2.5) 1 1.5 1.491 1.9261 0.1652217

s(BbMx.2.5.1) 1 0.2 0.170 0.2194 0.6395060

s(BbAv.2.5.1) 1 0.0 0.039 0.0508 0.8217513

s(BbAH) 1 1.0 1.045 1.3496 0.2453751

s(BbAHh) 1 11.1 11.054 14.2777 0.0001588 \*\*\*

s(BbMxAHH) 1 9.5 9.485 12.2515 0.0004672 \*\*\*

s(BbAvAHH) 1 2.0 2.022 2.6121 0.1060878

s(BbMxAHA) 1 0.0 0.028 0.0356 0.8503098

s(BbAvAHA) 1 3.3 3.317 4.2846 0.0384879 \*

s(PSCH) 1 49.1 49.113 63.4368 1.86e-15 \*\*\*

s(PSCD) 1 0.1 0.054 0.0692 0.7925662

s(PSCA) 1 9.0 8.951 11.5615 0.0006763 \*\*\*

Residuals 8871 6868.0 0.774

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Anova for Nonparametric Effects

Npar Df Npar F Pr(F)

(Intercept)

s(B365H) 3 0.3657 0.7777495

s(B365D) 3 0.3474 0.7910117

s(B365A) 3 0.2676 0.8487769

s(BWH) 3 0.8453 0.4689082

s(BWD) 3 0.4307 0.7310176

s(BWA) 3 1.5258 0.2055769

s(IWH) 3 3.5610 0.0136071 \*

s(IWD) 3 0.5743 0.6318719

s(IWA) 3 0.2200 0.8825302

s(LBH) 3 0.2706 0.8466239

s(LBD) 3 0.4878 0.6907338

s(LBA) 3 1.1158 0.3411633

s(PSH) 3 0.4581 0.7115874

s(PSD) 3 1.0126 0.3859002

s(PSA) 3 0.5212 0.6676732

s(WHH) 3 0.2222 0.8810178

s(WHD) 3 1.4717 0.2200765

s(WHA) 3 0.2369 0.8707204

s(VCH) 3 0.2078 0.8910620

s(VCD) 3 1.1816 0.3150497

s(VCA) 3 0.1867 0.9055193

s(Bb1X2) 3 1.9473 0.1196419

s(BbMxH) 3 0.1835 0.9076421

s(BbAvH) 3 0.2114 0.8885472

s(BbMxD) 3 0.2629 0.8521315

s(BbAvD) 3 0.4481 0.7186263

s(BbMxA) 3 0.5274 0.6634639

s(BbAvA) 3 0.2677 0.8487538

s(BbOU) 3 4.5067 0.0036531 \*\*

s(BbMx.2.5) 3 0.2221 0.8811050

s(BbAv.2.5) 3 0.2411 0.8676866

s(BbMx.2.5.1) 3 0.7657 0.5130877

s(BbAv.2.5.1) 3 0.2725 0.8452890

s(BbAH) 3 2.1700 0.0893414 .

s(BbAHh) 3 2.4457 0.0619622 .

s(BbMxAHH) 3 6.3647 0.0002639 \*\*\*

s(BbAvAHH) 3 1.4581 0.2238626

s(BbMxAHA) 3 0.4679 0.7047067

s(BbAvAHA) 3 0.6463 0.5851961

s(PSCH) 3 4.6676 0.0029148 \*\*

s(PSCD) 3 0.8632 0.4593584

s(PSCA) 3 0.4395 0.7247890

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> pred.gam1 <- predict(gam1, newdata = Football.test)

> mean((pred.gam1 - Football.test$HDCPOH)^2)

[1] 0.7724365

The test MSE is given by 0.772. Notice that when handicap bet on home team loses, the investor will either receive zero dividend (lose all) or half-refunded (partial lose); when handicap draw happens, the investor got their investment fully refunded; when handicap win happens, the investor will got the dividend at odds of around 1.35-1.6 (partial win) or around 1.65-2.2 (complete win). Thus, the test MSE here is considered large. This GAM fitting included the odds of all available bookmakers in the data, but it can be seen in the summary that quite a lot of variables are in fact, statistically insignificant that they are related to the home-team paying odd.

To interpret the data in pred.gam1, we *expect* that if the predicted home-paying odd is high than that of away team, we should buy home team for higher chance of winning. Now we will perform the same procedure to generate the prediction of away paying odds:

> set.seed(1)

> gam2 <- gam(HDCPOA ~ s(B365H) + s(B365D) + s(B365A) + s(BWH) + s(BWD) + s(BWA) + s(IWH) + s(IWD) + s(IWA) + s(LBH) + s(LBD) + s(LBA) + s(PSH) + s(PSD) + s(PSA) + s(WHH) + s(WHD) + s(WHA) + s(VCH) + s(VCD) + s(VCA) + s(Bb1X2) + s(BbMxH) + s(BbAvH) + s(BbMxD) + s(BbAvD) + s(BbMxA) + s(BbAvA) + s(BbOU) + s(BbMx.2.5) + s(BbAv.2.5) + s(BbMx.2.5.1) + s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) + s(BbMxAHH) + s(BbAvAHH) + s(BbMxAHA) + s(BbAvAHA) + s(PSCH) + s(PSCD) + s(PSCA), data = Football.train)

> summary(gam2)

Call: gam(formula = HDCPOA ~ s(B365H) + s(B365D) + s(B365A) + s(BWH) +

s(BWD) + s(BWA) + s(IWH) + s(IWD) + s(IWA) + s(LBH) + s(LBD) +

s(LBA) + s(PSH) + s(PSD) + s(PSA) + s(WHH) + s(WHD) + s(WHA) +

s(VCH) + s(VCD) + s(VCA) + s(Bb1X2) + s(BbMxH) + s(BbAvH) +

s(BbMxD) + s(BbAvD) + s(BbMxA) + s(BbAvA) + s(BbOU) + s(BbMx.2.5) +

s(BbAv.2.5) + s(BbMx.2.5.1) + s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) +

s(BbMxAHH) + s(BbAvAHH) + s(BbMxAHA) + s(BbAvAHA) + s(PSCH) +

s(PSCD) + s(PSCA), data = Football.train)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.52808 -0.92952 0.03533 0.89252 1.59702

(Dispersion Parameter for gaussian family taken to be 0.7709)

Null Deviance: 7002.932 on 9039 degrees of freedom

Residual Deviance: 6838.535 on 8870.998 degrees of freedom

AIC: 23471.48

431 observations deleted due to missingness

Number of Local Scoring Iterations: 23

Anova for Parametric Effects

Df Sum Sq Mean Sq F value Pr(>F)

s(B365H) 1 1.8 1.838 2.3839 0.1226301

s(B365D) 1 5.4 5.412 7.0203 0.0080733 \*\*

s(B365A) 1 6.9 6.873 8.9153 0.0028356 \*\*

s(BWH) 1 0.2 0.246 0.3185 0.5725095

s(BWD) 1 0.2 0.202 0.2621 0.6087183

s(BWA) 1 0.0 0.002 0.0025 0.9601924

s(IWH) 1 6.4 6.388 8.2865 0.0040036 \*\*

s(IWD) 1 4.3 4.299 5.5770 0.0182195 \*

s(IWA) 1 2.7 2.677 3.4731 0.0624077 .

s(LBH) 1 1.0 0.998 1.2946 0.2552371

s(LBD) 1 4.3 4.295 5.5709 0.0182824 \*

s(LBA) 1 0.0 0.042 0.0549 0.8147805

s(PSH) 1 6.0 5.979 7.7562 0.0053641 \*\*

s(PSD) 1 7.1 7.139 9.2606 0.0023483 \*\*

s(PSA) 1 0.0 0.003 0.0041 0.9490611

s(WHH) 1 3.9 3.868 5.0174 0.0251180 \*

s(WHD) 1 0.0 0.008 0.0100 0.9205202

s(WHA) 1 0.8 0.840 1.0901 0.2964781

s(VCH) 1 1.8 1.832 2.3763 0.1232231

s(VCD) 1 1.7 1.684 2.1841 0.1394783

s(VCA) 1 2.9 2.926 3.7958 0.0514114 .

s(Bb1X2) 1 1.1 1.128 1.4631 0.2264653

s(BbMxH) 1 5.6 5.603 7.2680 0.0070326 \*\*

s(BbAvH) 1 0.0 0.000 0.0000 0.9945969

s(BbMxD) 1 4.1 4.075 5.2866 0.0215130 \*

s(BbAvD) 1 0.9 0.855 1.1095 0.2922300

s(BbMxA) 1 0.3 0.295 0.3821 0.5364898

s(BbAvA) 1 0.6 0.573 0.7432 0.3886689

s(BbOU) 1 0.9 0.889 1.1526 0.2830367

s(BbMx.2.5) 1 0.2 0.163 0.2113 0.6457262

s(BbAv.2.5) 1 1.1 1.054 1.3667 0.2424166

s(BbMx.2.5.1) 1 0.7 0.693 0.8993 0.3429909

s(BbAv.2.5.1) 1 0.1 0.066 0.0850 0.7706093

s(BbAH) 1 1.0 0.964 1.2507 0.2634397

s(BbAHh) 1 11.9 11.872 15.4002 8.764e-05 \*\*\*

s(BbMxAHH) 1 9.4 9.444 12.2510 0.0004673 \*\*\*

s(BbAvAHH) 1 2.7 2.689 3.4880 0.0618464 .

s(BbMxAHA) 1 1.4 1.373 1.7807 0.1821008

s(BbAvAHA) 1 1.5 1.473 1.9104 0.1669571

s(PSCH) 1 48.6 48.557 62.9890 2.331e-15 \*\*\*

s(PSCD) 1 0.2 0.237 0.3074 0.5793096

s(PSCA) 1 8.9 8.919 11.5699 0.0006732 \*\*\*

Residuals 8871 6838.5 0.771

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Anova for Nonparametric Effects

Npar Df Npar F Pr(F)

(Intercept)

s(B365H) 3 0.3063 0.8208403

s(B365D) 3 0.4791 0.6968256

s(B365A) 3 0.3496 0.7894387

s(BWH) 3 0.8215 0.4818017

s(BWD) 3 0.5502 0.6480101

s(BWA) 3 1.3445 0.2579284

s(IWH) 3 3.3450 0.0183146 \*

s(IWD) 3 0.8128 0.4865778

s(IWA) 3 0.2156 0.8855992

s(LBH) 3 0.2807 0.8393744

s(LBD) 3 0.5798 0.6282432

s(LBA) 3 1.3466 0.2572753

s(PSH) 3 0.5041 0.6794391

s(PSD) 3 1.2366 0.2946180

s(PSA) 3 0.5342 0.6588068

s(WHH) 3 0.1717 0.9155781

s(WHD) 3 1.6364 0.1786491

s(WHA) 3 0.2787 0.8408579

s(VCH) 3 0.2074 0.8913241

s(VCD) 3 1.2055 0.3060321

s(VCA) 3 0.1837 0.9075500

s(Bb1X2) 3 1.8327 0.1388277

s(BbMxH) 3 0.2669 0.8492995

s(BbAvH) 3 0.2676 0.8488207

s(BbMxD) 3 0.2671 0.8491581

s(BbAvD) 3 0.5893 0.6219491

s(BbMxA) 3 0.5551 0.6446917

s(BbAvA) 3 0.3897 0.7604055

s(BbOU) 3 4.8477 0.0022639 \*\*

s(BbMx.2.5) 3 0.2852 0.8361197

s(BbAv.2.5) 3 0.1968 0.8986247

s(BbMx.2.5.1) 3 0.7181 0.5410633

s(BbAv.2.5.1) 3 0.2666 0.8495357

s(BbAH) 3 2.2617 0.0791442 .

s(BbAHh) 3 1.6698 0.1712013

s(BbMxAHH) 3 5.8619 0.0005398 \*\*\*

s(BbAvAHH) 3 2.1686 0.0895100 .

s(BbMxAHA) 3 0.6349 0.5924171

s(BbAvAHA) 3 1.0272 0.3792856

s(PSCH) 3 4.2521 0.0052142 \*\*

s(PSCD) 3 0.7227 0.5382563

s(PSCA) 3 0.3617 0.7806936

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

>

> pred.gam2 <- predict(gam2, newdata = Football.test)

> mean((pred.gam2 - Football.test$HDCPOA)^2)

[1] 0.7672059

Next, we compute the percentage of correct prediction:

> pred.CC1 <- rep(NA,nrow(Football.test))

> for (i in 1:nrow(Football.test)){

+ if (pred.gam1[i] < pred.gam2[i]){

+ pred.CC1[i] <- "A"

+ } else {

+ pred.CC1[i] <- "H"

+ }

+ }

> table(pred.CC1, Football.test$CC)

pred.CC1 A H

A 404 349

H 412 404

> 808/1569

[1] 0.5149777

The percentage of correct prediction is 51.5%, which is unsatisfactory as it is just like pure guessing. We suspect that we “thought too much”. That is, we have too many variables. Thus, we try the same method with fewer variables again:

> gam3 <- gam(HDCPOH ~ s(BbMxH) + s(BbAvH) + s(BbMxD) + s(BbAvD) + s(BbMxA) + s(BbAvA) + s(BbMx.2.5) + s(BbAv.2.5) + s(BbMx.2.5.1) + s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) + s(BbMxAHH) + s(BbAvAHH) + s(BbMxAHA) + s(BbAvAHA), data = Football.train)

> gam4 <- gam(HDCPOA ~ s(BbMxH) + s(BbAvH) + s(BbMxD) + s(BbAvD) + s(BbMxA) + s(BbAvA) + s(BbMx.2.5) + s(BbAv.2.5) + s(BbMx.2.5.1) + s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) + s(BbMxAHH) + s(BbAvAHH) + s(BbMxAHA) + s(BbAvAHA), data = Football.train)

> pred.gam3 <- predict(gam3, newdata = Football.test)

> pred.gam4 <- predict(gam4, newdata = Football.test)

> mean((pred.gam3 - Football.test$HDCPOH)^2)

[1] 0.7659426

> mean((pred.gam4 - Football.test$HDCPOA)^2)

[1] 0.7599577

> pred.CC2 <- rep(NA,nrow(Football.test))

> for (i in 1:nrow(Football.test)){

+ if (pred.gam3[i] < pred.gam4[i]){

+ pred.CC2[i] <- "A"

+ } else {

+ pred.CC2[i] <- "H"

+ }

+ }

> table(pred.CC2,Football.test$CC)

pred.CC2 A H

A 331 318

H 485 435

> (331+435)/1569

[1] 0.4882091

Although the test MSE slightly reduced in comparison to the fitting done with all available odds, the test percentage accuracy is still unsatisfactory. It is because GAM has a major disadvantage that under many variables, some important interactions can be missed.

Now the investigation with GAM is over, and we think it is time to switch to tree-based method. Still we shall first start with all available odds, then reduce the number of predictors as we move ahead. Firstly, we try the classification tree:

> library(tree)

> tree1 <- tree(CC ~ s(B365H) + s(B365D) + s(B365A) + s(BWH) + s(BWD) + s(BWA) + s(IWH) + s(IWD) + s(IWA) + s(LBH) + s(LBD) + s(LBA) + s(PSH) + s(PSD) + s(PSA) + s(WHH) + s(WHD) + s(WHA) + s(VCH) + s(VCD) + s(VCA) + s(Bb1X2) + s(BbMxH) + s(BbAvH) + s(BbMxD) + s(BbAvD) + s(BbMxA) + s(BbAvA) + s(BbOU) + s(BbMx.2.5) + s(BbAv.2.5) + s(BbMx.2.5.1) + s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) + s(BbMxAHH) + s(BbAvAHH) + s(BbMxAHA) + s(BbAvAHA) + s(PSCH) + s(PSCD) + s(PSCA), data = Football.train)

> summary(tree1)

Classification tree:

tree(formula = CC ~ s(B365H) + s(B365D) + s(B365A) + s(BWH) +

s(BWD) + s(BWA) + s(IWH) + s(IWD) + s(IWA) + s(LBH) + s(LBD) +

s(LBA) + s(PSH) + s(PSD) + s(PSA) + s(WHH) + s(WHD) + s(WHA) +

s(VCH) + s(VCD) + s(VCA) + s(Bb1X2) + s(BbMxH) + s(BbAvH) +

s(BbMxD) + s(BbAvD) + s(BbMxA) + s(BbAvA) + s(BbOU) + s(BbMx.2.5) +

s(BbAv.2.5) + s(BbMx.2.5.1) + s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) +

s(BbMxAHH) + s(BbAvAHH) + s(BbMxAHA) + s(BbAvAHA) + s(PSCH) +

s(PSCD) + s(PSCA), data = Football.train)

Variables actually used in tree construction:

character(0)

Number of terminal nodes: 1

Residual mean deviance: 1.386 = 12530 / 9039

Misclassification error rate: 0.4997 = 4517 / 9040

The misclassification error rate is about 50%, which is pretty close to pure guessing. We now try to fit another classification with less number of variables:

> tree2 <- tree(CC ~ s(BbMxH) + s(BbAvH) + s(BbMxD) + s(BbAvD) + s(BbMxA) + s(BbAvA) + s(BbMx.2.5) + s(BbAv.2.5) + s(BbMx.2.5.1) + s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) + s(BbMxAHH) + s(BbAvAHH) + s(BbMxAHA) + s(BbAvAHA), data = Football.train)

> summary(tree2)

Classification tree:

tree(formula = CC ~ s(BbMxH) + s(BbAvH) + s(BbMxD) + s(BbAvD) +

s(BbMxA) + s(BbAvA) + s(BbMx.2.5) + s(BbAv.2.5) + s(BbMx.2.5.1) +

s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) + s(BbMxAHH) + s(BbAvAHH) +

s(BbMxAHA) + s(BbAvAHA), data = Football.train)

Variables actually used in tree construction:

character(0)

Number of terminal nodes: 1

Residual mean deviance: 1.386 = 13130 / 9470

Misclassification error rate: 0.4988 = 4724 / 9471

We can see the improvement is negligible. Both fitted trees are single-node empty tree. Now we try to fit the regression trees for the home paying odds and away paying odds:

> rtree1 <- tree(HDCPOH ~ s(B365H) + s(B365D) + s(B365A) + s(BWH) + s(BWD) + s(BWA) + s(IWH) + s(IWD) + s(IWA) + s(LBH) + s(LBD) + s(LBA) + s(PSH) + s(PSD) + s(PSA) + s(WHH) + s(WHD) + s(WHA) + s(VCH) + s(VCD) + s(VCA) + s(Bb1X2) + s(BbMxH) + s(BbAvH) + s(BbMxD) + s(BbAvD) + s(BbMxA) + s(BbAvA) + s(BbOU) + s(BbMx.2.5) + s(BbAv.2.5) + s(BbMx.2.5.1) + s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) + s(BbMxAHH) + s(BbAvAHH) + s(BbMxAHA) + s(BbAvAHA) + s(PSCH) + s(PSCD) + s(PSCA), data = Football.train)

> summary(rtree1)

Regression tree:

tree(formula = HDCPOH ~ s(B365H) + s(B365D) + s(B365A) + s(BWH) +

s(BWD) + s(BWA) + s(IWH) + s(IWD) + s(IWA) + s(LBH) + s(LBD) +

s(LBA) + s(PSH) + s(PSD) + s(PSA) + s(WHH) + s(WHD) + s(WHA) +

s(VCH) + s(VCD) + s(VCA) + s(Bb1X2) + s(BbMxH) + s(BbAvH) +

s(BbMxD) + s(BbAvD) + s(BbMxA) + s(BbAvA) + s(BbOU) + s(BbMx.2.5) +

s(BbAv.2.5) + s(BbMx.2.5.1) + s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) +

s(BbMxAHH) + s(BbAvAHH) + s(BbMxAHA) + s(BbAvAHA) + s(PSCH) +

s(PSCD) + s(PSCA), data = Football.train)

Variables actually used in tree construction:

character(0)

Number of terminal nodes: 1

Residual mean deviance: 0.7776 = 7029 / 9039

Distribution of residuals:

Min. 1st Qu. Median Mean 3rd Qu. Max.

-0.97460 -0.97460 0.02537 0.00000 0.90540 1.52500

> rtree2 <- tree(HDCPOH ~ s(BbMxH) + s(BbAvH) + s(BbMxD) + s(BbAvD) + s(BbMxA) + s(BbAvA) + s(BbMx.2.5) + s(BbAv.2.5) + s(BbMx.2.5.1) + s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) + s(BbMxAHH) + s(BbAvAHH) + s(BbMxAHA) + s(BbAvAHA), data = Football.train)

> summary(rtree2)

Regression tree:

tree(formula = HDCPOH ~ s(BbMxH) + s(BbAvH) + s(BbMxD) + s(BbAvD) +

s(BbMxA) + s(BbAvA) + s(BbMx.2.5) + s(BbAv.2.5) + s(BbMx.2.5.1) +

s(BbAv.2.5.1) + s(BbAH) + s(BbAHh) + s(BbMxAHH) + s(BbAvAHH) +

s(BbMxAHA) + s(BbAvAHA), data = Football.train)

Variables actually used in tree construction:

character(0)

Number of terminal nodes: 1

Residual mean deviance: 0.7767 = 7355 / 9470

Distribution of residuals:

Min. 1st Qu. Median Mean 3rd Qu. Max.

-0.97250 -0.97250 0.02745 0.00000 0.90750 1.52700

Again, both trees fitted are empty. Next, we try random forests method. Firstly, we try the bagging to fit the data with tree method.

> bag1 <- randomForest(HDCPOH ~ B365H + B365D + B365A + BWH + BWD + BWA + IWH + IWD + IWA + LBH + LBD + LBA + PSH + PSD + PSA + WHH + WHD + WHA + VCH + VCD + VCA + Bb1X2 + BbMxH + BbAvH + BbMxD + BbAvD + BbMxA + BbAvA + BbOU + BbMx.2.5 + BbAv.2.5 + BbMx.2.5.1 + BbAv.2.5.1 + BbAH + BbAHh + BbMxAHH + BbAvAHH + BbMxAHA + BbAvAHA + PSCH + PSCD + PSCA, data = Football.train, mtry = 42, importance = TRUE)

It takes around 10 minutes for R to finish the fitting. However, the importance() function says none of the variables shows significant importance to the bagging fitting. Anyway, we will fit HDCPOA using bagging with similar manner:

bag2 <- randomForest(HDCPOA ~ B365H + B365D + B365A + BWH + BWD + BWA + IWH + IWD + IWA + LBH + LBD + LBA + PSH + PSD + PSA + WHH + WHD + WHA + VCH + VCD + VCA + Bb1X2 + BbMxH + BbAvH + BbMxD + BbAvD + BbMxA + BbAvA + BbOU + BbMx.2.5 + BbAv.2.5 + BbMx.2.5.1 + BbAv.2.5.1 + BbAH + BbAHh + BbMxAHH + BbAvAHH + BbMxAHA + BbAvAHA + PSCH + PSCD + PSCA, data = Football.train, mtry = 42, importance = TRUE)

Then we draw the confusion matrix of the classification correctness:

> pred.bag1 <- predict(bag1,newdata = Football.test)

> pred.bag2 <- predict(bag2, newdata = Football.test)

> pred.CC3 <- rep(NA,nrow(Football.test))

> for (i in 1:nrow(Football.test)){

+ if (pred.bag1[i] < pred.bag2[i]){

+ pred.CC3[i] <- "A"

+ } else {

+ pred.CC3[i] <- "H"

+ }

+ }

> table(pred.CC3, Football.test$CC)

pred.CC3 A H

A 395 376

H 421 377

>

> (395+377)/1569

[1] 0.4920331

The prediction correctness using bagging of predicted home/away odds are still close to pure guessing. We will now try fewer number of variables but larger number of trees. We use parallel computing to accelerate our random forest fitting:

> library(parallel)

> library(foreach)

> library(doSNOW)

Loading required package: iterators

Loading required package: snow

Attaching package: ‘snow’

The following objects are masked from ‘package:parallel’:

clusterApply, clusterApplyLB,

clusterCall, clusterEvalQ,

clusterExport, clusterMap,

clusterSplit, makeCluster, parApply,

parCapply, parLapply, parRapply,

parSapply, splitIndices, stopCluster

> library(snow)

> library(snowfall)

> library(randomForest)

randomForest 4.6-14

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

> registerDoSNOW(makeCluster(8, type = "SOCK"))

> set.seed(123)

> cindex.hdc <- 57:61

> train.hdc.x <- Football.train[,cindex.hdc]

> train.hdc.HDCPOH <- Football.train[,62]

> train.hdc.HDCPOA <- Football.train[,63]

> bag5 <- foreach(ntree = rep(1000,8), .combine = combine, .packages = "randomForest") %dopar%

+ randomForest(train.hdc.x, train.hdc.HDCPOH, ntree = ntree)

> bag6 <- foreach(ntree = rep(1000,8), .combine = combine, .packages = "randomForest") %dopar%

+ randomForest(train.hdc.x, train.hdc.HDCPOA, ntree = ntree)

It takes about 3 minutes to fit a random forest of 8000 trees for each bagging fitting. Next, we do the prediction of odds on testing data, and run the for loop for classification of data:

> library(stats)

> install.packages("itertools")

Installing package into ‘C:/Users/Globas Wu/Documents/R/win-library/3.4’

(as ‘lib’ is unspecified)

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/itertools\_0.1-3.zip'

Content type 'application/zip' length 79901 bytes (78 KB)

downloaded 78 KB

package ‘itertools’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Globas Wu\AppData\Local\Temp\Rtmp4O2UUc\downloaded\_packages

> library(itertools)

> pred.bag5 <- foreach(d = isplitRows(Football.test, chunks = 8), .combine = c, .packages = "stats") %dopar% {

+ predict(bag5, newdata = d)

+ }

> pred.bag6 <- foreach(d = isplitRows(Football.test, chunks = 8), .combine = c, .packages = "stats") %dopar% {

+ predict(bag6, newdata = d)

+ }

> pred.CC4 <- rep(NA, nrow(Football.test))

> for (i in 1:nrow(Football.test)){

+ if (pred.bag5[i] < pred.bag6[i]){

+ pred.CC4[i] <- "A"

+ } else {

+ pred.CC4[i] <- "H"

+ }

+ }

> table(pred.CC4, Football.test$CC)

pred.CC4 A H

A 382 370

H 434 383

> 765/1569

[1] 0.4875717

The percentage of correct prediction is still close to merely guessing. We suspect overfitting occurred in this case due to the high number of trees used. However, with a lower number of trees (800) applied to the training data, the test error rate has no significant improvement. We believe this will repeat no matter which other methods we would use.

**Conclusion**

In this project, we tried to attack the Football data by using smooth splines and random forest on odds predictors. The results seem unsatisfactory as in most cases the classification error is close to 50%, which is a value just like mere guessing. We will try to add some more predictors, such as league table data of previous years, in the upcoming final project and hopefully those newly-added predictors can play important role in classification of Asian handicap betting results.